

A Hybrid Metaheuristic Approach to Optimize the Districting Design of a Parcel Company

R.G. González-Ramírez^{*1}, N. R. Smith², R. G. Askin³, Pablo A. Miranda⁴, J.M. Sánchez⁵

^{1,4} Pontificia Universidad Católica de Valparaíso,
Escuela de Ingeniería Industrial
Av. Brasil 2251, Valparaíso, Chile.
*rosa.gonzalez@ucv.cl

^{2,5} ITESM, Campus Monterrey, Centro de Calidad y Manufactura.
2501 Eugenio Garza Sada South Av. Monterrey,
Nuevo León, Mexico. CP. 64849.

³Arizona State University. School of Computing,
Informatics and Decision Systems Engineering,
Arizona State University, Tempe, AZ 85287-8809, USA

ABSTRACT

In this article we address a districting problem faced by a pickup and delivery parcel company over a determined service region. The service region is divided into districts, each served by a single vehicle that departs from a central depot. Two objectives are optimized: compactness and balance of the workload content among the districts. We present a mathematical formulation of the problem and a heuristic algorithm to solve the problem. Numerical results are presented in comparison to CPLEX 11.1 solutions for the smaller size instances. The results show that the heuristic performs well. The algorithm is able to solve moderate size instances in reasonable computational time, given the strategic nature of the problem.

Keywords: Districting, hybrid algorithm, metaheuristic, graph, Tabu search, GRASP.

RESUMEN

En este artículo abordamos un problema de sectorización para actividades logísticas que enfrenta una empresa de entrega y recolección de paquetería en una determinada región. La región a la que se da servicio es dividida en distritos, cada uno atendido por un solo vehículo que sale de un depósito central. Se busca optimizar dos objetivos: compactidad y balance de la carga de trabajo. Presentamos la formulación matemática del problema y un algoritmo heurístico para resolverlo. Se presentan resultados numéricos en comparación con las soluciones encontradas por CPLEX 11.1 para las instancias de menor tamaño. Los resultados muestran un buen desempeño del algoritmo heurístico, el cual es capaz de resolver instancias de tamaño moderado en tiempos razonables, dada la naturaleza estratégica del problema.

1. Introduction

Districting problems involve the partitioning of a region into smaller size areas in order to facilitate the operations performed over the region and optimize determined criteria under consideration. Districting problems arise in a broad range of real applications such as politics, sales territory alignment, schools, health care systems, emergency sites, and logistics, where it is commonly associated with routing activities.

In this article we address a logistics districting problem motivated by a real-world application from

a parcel company that serves the metropolitan region of Monterrey, Mexico. The operations of the company involve the delivery and pickup of packages within a region. The region is divided into “districts” (zones), each served by a single vehicle that departs from a central depot. The districting process aims to optimize two criteria: compactness and workload balance.

Although the exact location of the customers and the daily volume of demand are stochastic, the problem is solved using a deterministic model. The procedure employed by the company to design districts is the following: Using the data of a

representative day, the locations of the customers are identified on a map and the districts are defined following what the company calls a spiral-sweep procedure in which they attempt to balance the workload of the districts and create districts of compact shape. This procedure takes about three weeks to be completed and only makes adjustments to the current district design.

Figure 1 presents an example of a districting configuration. Customers are located over a plane in a geographical region with the depot located at the origin. Five districts are defined. The criteria to optimize are measured as follows: workload is estimated as the time to perform all the pickups and deliveries to perform within the district as well as an estimated time to travel from the depot to the district. The latter is determined as the travel time from the depot to the farthest point of the district. In Figure 1 we can observe that districts 1 and 4 are farther from the depot than districts 5, 2, and 3. Thus more time is required to reach those districts and the optimal configuration may assign less workload to those districts in order to balance workload among them, depending on the time required to perform each service compared to the travel time within the district. Compactness has to do with the form of the district, which is desired to be as round or rectangular as possible with a high work density. We estimate compactness as the farthest distance between two points of a district. If

we assume Euclidean distances, we can observe that districts 1, 2, and 5 are more compact than the rest of the districts and district 3 would have the biggest travel time between its farthest points.

The objective of this work is to propose a methodology that is faster and allows the company to find a new and optimized district design. The methodology proposed is based on a heuristic algorithm, which is justified by the difficulty of the problem that has been shown to be NP -Complete in (Altman, 1997), reason for which an exact method may not be practical.

The remainder of the paper is organized as follows. Section 2 presents a literature review. Section 3 presents the mathematical model of the problem. Section 4 describes the solution methodology and section 5 presents the numerical results. Conclusions and recommendations for future research are shown in section 6.

2. Literature review

Several articles related to the districting problem in its different contexts can be found in the literature. To the best of our knowledge, the specific characteristics of the problem addressed here makes it unique and not previously addressed, however, we can find several works related to logistics districting problems that are similar. For

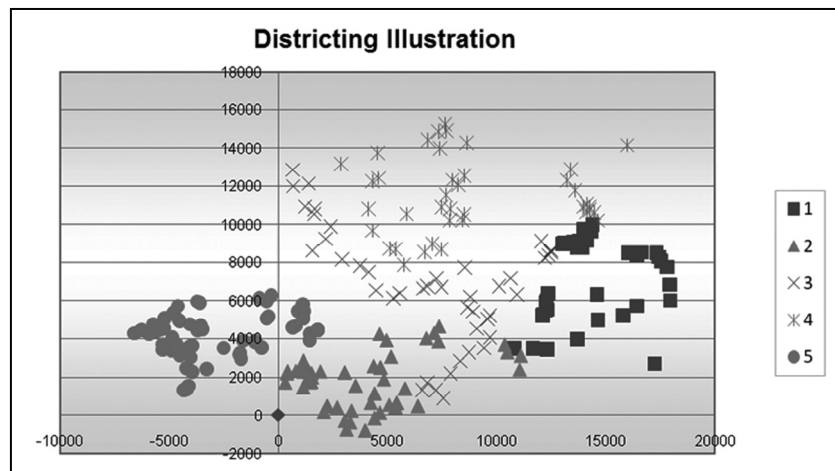


Figure 1. Districting Illustration.

instance, Langevin and Soumis (1989) study the problem of designing multiple vehicle delivery tours satisfying time constraints for the letter and parcel pickup and delivery problem, considering that the depot is located at the center of the service area and demand is randomly distributed with a density as a function of the radius, assuming a ring-radial grid. Our model differs in that we model the problem as a graph.

Also related is the work by Novaes and Gracioli (1999), who present a methodology to design multi-delivery tours over an urban region of irregular shape in which the density of demand points and the workload content varies over the service region. Novaes et al. (2000) extend the previous work with a continuous approach. Muyldermans et al. (2002, 2003) address the districting design problem for arc routing operations for salt spreading on winter roads. The authors present a graph model in which demand occurs at the edges. This differs from our model, in which demand occurs at each vertex. In the former work, the authors propose a heuristic procedure in which the total area is partitioned into cycles which are subsequently aggregated in two phases. In the latter work, the authors propose three different heuristics and then compare the solution values with a multi-depot CARP cutting plane lower bound. Haugland et al. (2007) consider the problem of designing districts for vehicle routing problems with stochastic demands. They propose two procedures to solve the problem. The first method is based on a Tabu

Search (TS) approach and the second on a multistart approach. The authors propose a two-stage stochastic model with recourse with the aim to minimize travel times. Our work differs in that we propose a hybrid algorithm that combines elements of both types of metaheuristics and in that we strive to balance workload content. For an extensive and more general literature review, refer to Moonen (2004).

3. Mathematical model

Consider a connected, undirected graph $G(V,E)$ where V is the vertex set and E the edge set. The graph is generally not complete. All the edges $e_{rs} = (v_r, v_s)$ have a positive length and represent a real road between adjacent points v_r and v_s . Distances between points are edge lengths for those points that are connected in the graph and shortest path distances for other pairs of points. A district is defined as a subset of the points. Each vertex may require either a pickup or a delivery.

The aim of the districting procedure is to optimize two criteria: balance of the workload content among the districts and compactness of district shapes. Figure 2 shows a Graph G , in which V is the set of 20 points that are located on the map, and E is the set of edges (20 for this instance).

We propose a hierarchical optimization approach in which the first model to be solved is a linear programming problem in which the weighted sum of the compactness metric and the maximum

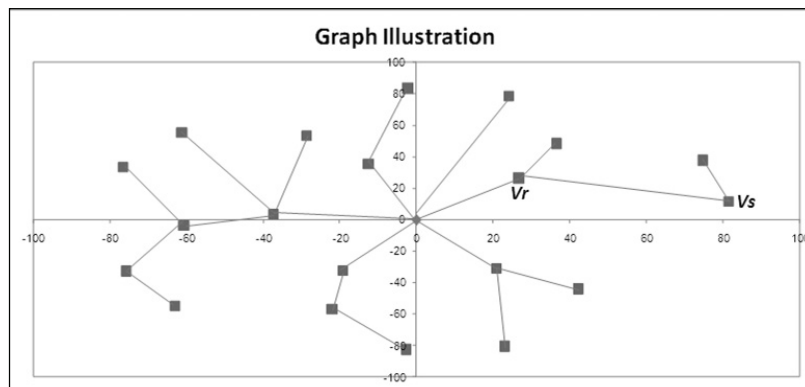


Figure 2. Graph Illustration.

workload content assigned to a district, each of them normalized, is minimized. Then a second optimization model is solved with the objective to minimize the dispersion of the workload content among the districts with the constraint that the previous objective may not be made worse.

The following notation is defined: α and β are the limits for the maximum number of pickups/deliveries for each district; $J = \{1, \dots, m\}$ is the district set; wp_i and wd_i are the number of pickups/deliveries requested by demand point i ; Stp and Std are the stopping time per pickups/delivery in each demand point i , $i \in V$; d_{ik} is the distance from point i to point k ; d_{0l} is the distance from the depot to the point l ; λ is the weighting factor, $0 \leq \lambda \leq 1$; Sp is the average speed; and Nz and Nw are the normalization parameters

for the compactness and workload metrics, respectively.

The following variables are defined: W is a continuous variable that represents the maximum workload content assigned to a district, Z is a continuous variable that measures the compactness as the maximum travel time between the furthest apart points of a district, D_j is a continuous variable that takes the value of the traveling time from the depot to the farthest point of district j , M_j takes the value of the traveling time between the two furthest apart points of district j , $dispersion_j$ is a continuous auxiliary variable that takes the absolute value of the difference between the workload content of each district with respect to the average workload content assigned to the districts, and X_{ij} takes the value of 1 if customer i is assigned to district j and zero otherwise.

The first optimization model is as follows:

$$\text{Min } OF1 = \lambda W / Nw + (1 - \lambda)Z / Nz \quad (1)$$

subject to

$$\sum_{j \in J} X_{ij} = 1 \quad \forall i \in V \quad (2)$$

$$\sum_{i \in V} wp_i X_{ij} \leq \alpha \quad \forall j \in J \quad (3)$$

$$\sum_{i \in V} wd_i X_{ij} \leq \beta \quad \forall j \in J \quad (4)$$

$$D_j \geq d_{i0} X_{ij} \quad \forall j \in J, i \in V \quad (5)$$

$$Z \geq d_{ik} (X_{ij} + X_{kj} - 1) / Sp \quad \forall j \in J, i \in I, k \neq i \in V \quad (6)$$

$$W \geq Std \sum_{i \in V} wd_i X_{ij} + Stp \sum_{i \in V} wp_i X_{ij} + D_j / Sp \quad \forall j \in J \quad (7)$$

$$X_{ij} \in \{1, 0\} \quad \forall j \in J, i \in V \quad (8)$$

Equation (1) is the objective function that minimizes a weighted average of the maximum workload and maximum compactness metrics. The objectives are normalized and the relative weighting is given by λ . Constraints (2) guarantee that each demand point is assigned to exactly one district. Constraints (3) and (4) guarantee that each district has a maximum of α pickups and β deliveries, respectively. Constraints (5) guarantee that D_j takes the value of the time from the depot to the farthest point of each district j . Compactness is defined as the distance between the two furthest apart points in a district and we attempt to obtain compact districts by minimizing the maximum compactness metric, as it is guaranteed by

constraints (6). If points i and k are assigned to the same district, then both X_{ij} and X_{kj} will take a positive value and hence the constraint will be activated, guaranteeing that Z takes the value of the maximum travel time between a pair of points in a district. The workload content of a district is defined as the time required to perform all required pickups and deliveries and the time needed to drive from the depot to the farthest point in the district. In order to balance the workload content among districts, we minimize the maximum workload allocated to a district and constraints (7) guarantee that W takes the value of the maximum workload from among the districts. Constraints (8) are the binary requirements.

The second optimization model is as follows:

$$\text{Min } OF2 = \sum_{j \in J} dispersion_j \quad (9)$$

subject to

constraints (2) through (7)

$$D_j = \sum_{i \in V} d_{i0} (X_{ij} + Y_{ij} - 1) \quad \forall j \in J, \quad (10)$$

$$\sum_{i \in V} (X_{ij} + Y_{ij} - 1) = 1 \quad \forall j \in J, \quad (11)$$

$$X_{ij} + Y_{ij} \geq 1 \quad \forall i \in V, \forall j \in J, \quad (12)$$

$$dispersion_j \geq \frac{\sum_{i \in V} Std_i \cdot wd_i X_{ij} + \sum_{i \in V} Stp_i \cdot wp_i \cdot X_{ij} + D_j / Sp - \sum_{k \in J} \sum_{i \in V} Std_i \cdot wd_i \cdot X_{ik} + \sum_{i \in V} Stp_i \cdot wp_i \cdot X_{ik} + D_k / Sp}{|J|} \quad \forall j \in J, \quad (13)$$

$$dispersion_j \geq \frac{\sum_{k \in J} \sum_{i \in V} Std_i \cdot wd_i \cdot X_{ik} + \sum_{i \in V} Stp_i \cdot wp_i \cdot X_{ik} + D_k / Sp}{|J|} - \frac{\sum_{i \in V} Std_i \cdot wd_i X_{ij} + \sum_{i \in V} Stp_i \cdot wp_i \cdot X_{ij} + D_j / Sp}{|J|} \quad \forall j \in J, \quad (14)$$

$$\frac{\lambda W}{Nw} + \frac{(1-\lambda)Z}{Nz} \leq OF1 + \varepsilon \quad (15)$$

$$X_{ij}, Y_{ij} \in \{1, 0\} \quad \forall i \in V, \forall j \in J \quad (16)$$

Constraints (9) correspond to the objective function of the second optimization model that minimizes the sum of the absolute value of the dispersion of the workload content of each district with respect to the average workload. Constraints (10) to (12) are auxiliary constraints to guarantee that the variable D_j takes the exact value of the time to travel from the depot to the farthest point of each district. Constraints (13) and (14) guarantee that for each district j , the difference between its workload content and the average workload content takes a positive value. Constraint (15) is included in the model in order to guarantee that the objective function value of the first optimization model may be equal or less to the value obtained when this model was solved, plus an epsilon to avoid conflicts with numerical precision. Constraints (16) are the binary requirements.

Normalization parameters are estimated with respect to the optimum value. Nw is determined by estimating an average workload per district when it is approximately balanced. To determine the approximate area of the service region, it is assumed to be approximately of circular shape with the depot at the center. Nz estimates the length of the diameter (in time units) of a district, assuming the area (of the circular approximation) of the service region is divided equally into m identical circular districts.

4. Hybrid districting algorithm (HDA) description

In this section we present the details of the proposed hybrid districting algorithm that combines elements of GRASP and Tabu Search (TS). TS is an adaptive memory-based technique proposed in 1977 by Glover (1977) that enhances the performance of a local search procedure through the use of memory structures to aid escaping from local optima by accepting even non-improving moves. Comprehensive tutorials on Tabu Search are found in Glover and Laguna (1997, 2002).

GRASP is a multi-start constructive metaheuristic proposed by Feo and Resende (1989) in which an iteration consists of two phases: construction of an initial solution from scratch and then improvement of the solution by a local search approach. For a detailed description of GRASP, see Resende and Ribeiro (2002). Several initial solutions are constructed and improved. The best final solution is selected as the final districting configuration. Details of each phase are further described below.

4.1 Feasible initial solution construction (fISC)

We first select a set of seeds, one for each district. Then we allocate the remaining points to the districts associated to each seed. Constructing a feasible solution turns out to be a difficult task because of the requirement of satisfying two capacity limits of the districts. We may actually find some instances for which it is not possible to find a feasible solution.

To select the set of seeds we propose five different procedures that are employed by the algorithm in order to construct a feasible solution, which is then improved. The first four procedures select the seeds in a greedy randomized fashion. The first greedy function selects the seeds as those points that are most disperse to each other, based on a modified version of the greedy algorithm of Erkut (1991) that solves the p -dispersion problem. The second greedy function evaluates the points with respect to the number of "neighbors" located within a determined distance from each point and selects as seeds those points with a maximum number of neighbors. Each time a seed is selected, its neighbors are discarded as potential points. The third greedy function selects the seeds according to the location on the plane. The region is divided like a pie into equal sectors and those points located close to the corresponding division of each district are selected as candidates. Potential points are evaluated according to the greedy

function in use and the better choices are placed on a restricted candidate list (RCL) from which the seed is randomly selected. The fourth method is similar but it considers the workload content of the points to divide the region instead of just considering the location of the points. The last method is a semi-random approach that sequentially selects seed randomly. Each time a seed is selected, its neighbors are discarded to avoid selecting as seeds points that are very close to each other. In case that no potential point is available to select a seed, the seed is selected randomly among those points that were previously discarded.

Once a set of seeds has been selected, the remaining points are allocated to the districts, striving to find a feasible solution and enhance compactness. For this, a point is allocated to a district only if it is adjacent to any of the points already allocated to the district. Two districts are adjacent if at least one point in one district is connected to at least one point in the other district. We first try to make a feasible assignment of points (FAP), which means that we respect the capacity limits of the points. If required, points are assigned even if capacity limits are not respected, an infeasible assignment of points (IAP), and then a reallocation procedure is performed with the aim of obtaining a feasible solution (H-R). Details of the procedure are as follows:

Feasible assignment of points (FAP)

FAP.1 Select a seed randomly.

FAP.1.1 The closest unassigned point to the seed is selected. If the point is adjacent and there is enough capacity in the district, it is allocated to the district, otherwise the point is discarded. This procedure is repeated until a point has been allocated or a determined number of points has been explored.

FAP.1.2 Update adjacency relations between points and districts

Repeat the procedure during a determined number of iterations or until all points have been assigned to a district. If unassigned points remain, go to FAP.2, otherwise, stop the procedure.

FAP.2. Select an unassigned point sequentially

FAP.2.1 Select the district with the closest seed to the point and assign the point if there is enough capacity and the point is adjacent to the district.

FAP.2.2 Update adjacency relations between points and districts.

Repeat until all unassigned points have been explored with the aim of allocating them to a district. If unassigned points remain, go to IAP.1, otherwise, stop the procedure.

Infeasible assignment of points (IAP)

IP.1 Select an unassigned point sequentially.

IAP.1.1 Select a district, searching those districts with more available capacity first. If the selected point is adjacent to the district, the point is allocated even if capacity limits are not respected.

IAP.1.2 Update adjacency relations between points and districts.

The procedure is repeated until all points have been assigned.

IAP.2 Apply the hyperheuristic-reallocation procedure to get feasibility.

Hyperheuristic-reallocation procedure (H-R)

This procedure consists of reallocating points between adjacent districts with the aim of finding a feasible solution. We propose a simple hyperheuristic to guide the reallocation procedure. The heuristics attempt to find a pair of adjacent districts such that a point can be reassigned from one (the *sending* district) to the other (the *receiving* district). We propose six low level heuristics or decision rules, three of which allow only feasible moves. This means that a district can receive a point only if it has enough capacity. A TS list function is implemented that aids in escaping local optima. The procedure is repeated during a determined number of iterations or until a feasible solution has been constructed.

4.2 Local search (LS)

The procedure considers a search space that consists of the solutions found once a move of a point between a pair of adjacent districts is performed with the aim of finding a better districting configuration. A TS short term memory is

implemented with an aspiration criterion that allows a *tabu active* move only if the resulting solution is better than the current best solution. An iteration of the LS phase consists of the evaluation of all the solutions in the defined search space. The best solution found, as per Equation (1), is selected even if it is worse than the current best solution found. In case of ties, solutions are evaluated using Equation (25) and the solution that presents the least dispersion of workload content among districts is preferred.

$$WDispersion = \sum_{j \in J} |W_j - \bar{W}| \quad (17)$$

where

$$W_j = Std \sum_{i \in I'} wd_i X_{ij} + Stp \sum_{i \in I'} wp_i X_{ij} + D_j / Sp, \quad (18)$$

$$\bar{W} = \sum_{j \in J} W_j / |J| \quad (19)$$

The neighborhood structure consists of an evaluation of all the possible moves or transfers between adjacent districts including infeasible moves. This part of the procedure is referred to as the *first step*. For each solution found during the first step, now the search space consists of all the feasible solutions that result from moving a point between adjacent districts. This is referred to as the second step. Among all the solutions evaluated in the *second step*, the best solution is selected but only the move done during the first step is performed. If it turns out that it is an infeasible solution, then the move selected during the second step is also performed so that we always end up with a feasible solution.

Also, in case of ties, the solution that results in the lower dispersion metric according to Equation (17) is selected. A list of the three best solutions is maintained and at the end of the procedure a final attempt is made to improve these three solutions, but the search is done by evaluations of a single step. The overall best solution found is reported as the final solution for the given initial feasible solution.

5. Numerical results

To test the performance of the proposed solution procedure, a set of instances was generated. All problem instances were solved on a 2.00 GHz Pentium processor with 2 GB of RAM running under Windows XP. The instances generated were solved by the procedure proposed in this work and by CPLEX 11.1 (<http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>) for comparison purposes.

5.1 Instance generation

We defined two types of instances. The first type of instances is referred to as "*General Instances*", which are generated so as to resemble the structure of the metropolitan region of Monterrey. We generated a set of *base points* using Google Earth, taking care to locate the points where there actually exists a home, office, or building that may be visited by the vehicles. Instances are generated by randomly selecting a subset of points from the set of base points. The depot is located in the same place as it is for the parcel company, which is approximately at the center of the geographical region. Each point is assigned a pickup or a delivery at random with equal probability. The second type of instance is referred to as "*Symmetric Instances*", which are not realistic instances, but generated for comparison purposes. This type of instance is generated such that the optimal solution corresponds to symmetric districts with the same workload content and compactness metric. Euclidean distances are used to estimate distances between each pair of points (whether or not there exists an edge connecting the points).

For all instances of both types, the stopping times for the pickup and delivery activities were fixed at a realistic value suggested by the parcel company: 5 minutes for a delivery and 10 minutes for a pickup. We define three levels of average speed: 25 kilometers/hour, 30 kilometers/hour and 35 kilometers/hour. The limits on the number of pickups and deliveries are defined based on two levels of capacity: tight (T) and less restricted (LR), which are based on an average number of pickups/deliveries that would correspond to each district if the total number of services were divided

equally among the districts plus a tolerance term. For the tight instances we define it as the floor of 0.15 times the total number of points divided by the number of districts. For the less restricted instances we multiply the same amount by 0.22. We consider three instance sizes: 50 points/5 districts (50_5), 200 points/10 districts (200_10), and 450 points/15 districts (450_15). The weighting factor λ was set to three different values: 0.25, 0.5 and 0.75. For each of the two instance types, and the three instance sizes, three replicates were generated, and each of them were tested varying over the two capacity limits, the three values of the relative weighting factor as well as the three values of speed, resulting in total $2 \times 3 \times 3 \times 2 \times 3 \times 3 = 324$ instances.

5.2 Stopping rules

A limit time of 3600 seconds was set for the instances solved by CPLEX 11.1 for each optimization models consuming in total up to 7200 seconds (CPLEX 11.1 did not make a significant improvement beyond that). For the heuristic, a maximum limit time of 3600 seconds was set. We define a maximum number of iterations. For the Hyperheuristic-Reallocation procedure we set a maximum of 15 iterations for the 50-5 instances and 25 for the rest of the instances. For the LS procedure we set a maximum of 40 iterations. We also established a maximum of 25 attempts to construct an initial feasible solution, sequentially applying the five proposed methods to select a set of seeds.

5.3 Results

All instances were attempted to be solved by both the heuristic procedure and CPLEX 11.1, but CPLEX 11.1 could not find an integer solution for the 450_15 instances. For the rest of the instances we report the optimal or the best integer solution found by CPLEX 11.1. We compute a gap between the heuristic and CPLEX 11.1 solutions with respect to the *OF1* value reported by CPLEX 11.1 once the second optimization model is solved as defined by Equation (20). Positive gaps are obtained when CPLEX 11.1 finds a better solution than our procedure:

$$Gap = \frac{OF1 - CPLEX}{CPLEX} \tag{20}$$

For the symmetric instances, we compared the heuristic and CPLEX 11.1 solutions against the optimal solution which consists of symmetric districts with the same workload content and compactness metric. We first present the results of the 50_5 and 200_10 size instances in Tables 1 and 2, respectively, for which CPLEX 11.1 could find either the optimal or at least a feasible solution (with integer values of the decision variables X_{ij}). The minimum, average, and maximum gap between the heuristic solution (HDA) and CPLEX 11.1 solution are reported for the general instances, and gaps between the optimal solution and the heuristic and CPLEX 11.1 solutions for the

Size	Metric	URBAN INSTANCES						
		GAP (CPX-HDA)	OF1 CPX	OF1 HDA	OF2 CPX	OF2 HDA	Time CPX	Time HDA
50_5	Maximum	0.00	0.94	0.94	764.61	784.74	1174.66	10.06
	Average	0.00	0.74	0.74	584.92	589.74	467.12	6.59
	Minimum	0.00	0.54	0.54	445.11	439.56	300.70	4.34
200_10	Maximum	-0.30	1.99	0.92	739.48	1869.23	7209.07	406.16
	Average	-0.51	1.60	0.75	136.03	1284.90	7206.23	237.32
	Minimum	-0.71	1.26	0.54	59.36	914.91	7204.82	62.19

Table 1. Numerical results- General Instances, 50_5 and 200_10 size.

Size	Metric	SYMMETRIC INSTANCES							
		GAP (CPX -OPT)	GAP (HDA-OPT)	OF1 CPX	OF1 HDA	OF2 CPX	OF2 HAD	Time CPX	Time HDA
50_5	Maximum	0.00	0.00	0.95	0.95	0.00	0.33	621.97	3.97
	Average	0.00	0.00	0.87	0.87	0.00	0.06	84.94	2.90
	Minimum	0.00	0.00	0.74	0.74	0.00	0.00	6.16	2.05
200_10	Maximum	1.55	0.22	2.26	1.12	16.18	0.33	7210.09	258.91
	Average	0.93	0.05	1.71	0.98	10.40	0.21	7207.53	160.31
	Minimum	0.37	0.00	1.19	0.87	0.00	0.00	7205.47	77.72

Table 2. Numerical results- Symmetric Instances, 50_5 and 200_10 size.

symmetric instances. Also, computational times for both CPLEX 11.1 and the heuristic are reported, as well as the minimum, average, and maximum objective function (*OF1* and *OF2*) values. We would like to point out that the primary objective to optimize is *OF1*. The value reported for *OF1* is the value obtained after solving the second optimization model, and the value of *OF2* is equal to the absolute value of the dispersion of workload content, which is a secondary objective.

In general, we can observe from the tables that the heuristic performs very well. CPLEX 11.1 found the optimal solution for both types (general and symmetric) of the 50_5 instances. For these same instances, our procedure also found optimal solutions for both the general and symmetric instances. Regarding the computational time of this instance size, CPLEX 11.1 had a maximum computational time of 1174 seconds for the general instances and

621 seconds for the symmetric instances (which are easier to solve). The heuristic procedure reported very low computational times: 10.06 and 3.97 seconds, respectively.

In order to better observe the differences between the solutions found by CPLEX 11.1 and the heuristic, Figure 3 and 4 present a comparative chart for the general and symmetric instances, respectively. The primary axis (left side) refers to the values of the objective functions and the secondary axis (right side) to the computational times. As can be seen, CPLEX 11.1 and the heuristic both found the optimal solution for the general and symmetric instances, which is the reason why these values overlap. The only difference that we can appreciate in Figures 3 and 4 are in the computational times, with the heuristic always reporting lower times than CPLEX 11.1.

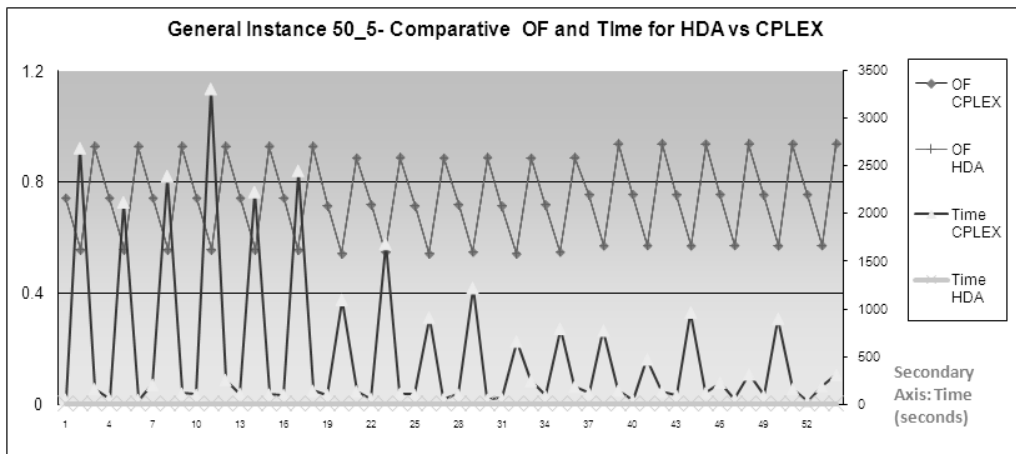


Figure 3. Comparative Chart, General Instances of size 50_5.

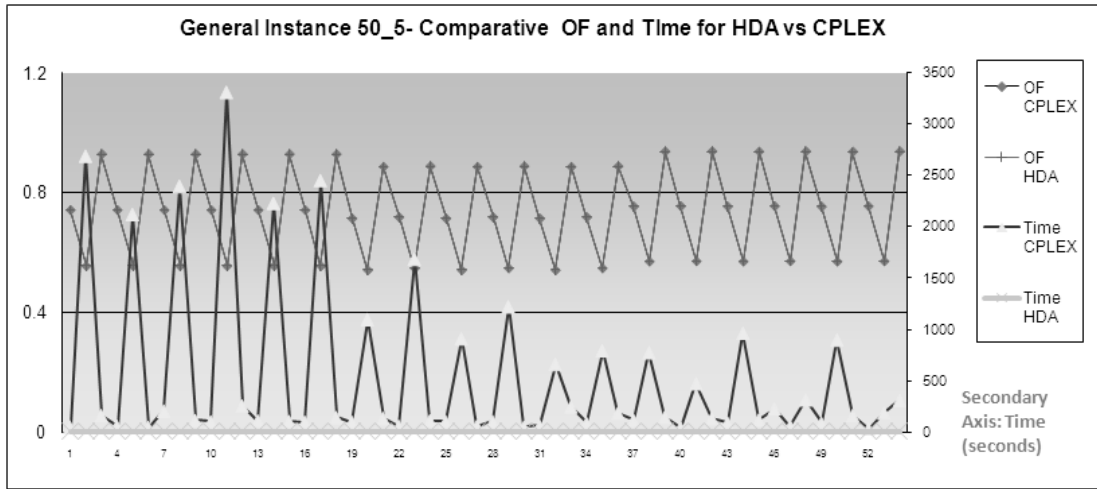


Figure 4. Comparative Chart, General Instances of size 50_5.

In the case of the 200_10 instances, CPLEX 11.1 could not find the optimal solution during the maximum computational time that was set, so we report in Tables 5.1 and 5.2 the best feasible (integer) solution that was found. Comparative charts for each instance type are shown in Figures 5 and 6. We can observe in Table 1 that negative gaps are reported. These indicate that the heuristic found a better solution than the solution reported

by CPLEX 11.1. These results are also shown in Figure 5, with the heuristic always achieving lower values of the objective function, and CPLEX 11.1 having the maximum computational times for the first optimization model and in most of the cases for the second optimization model. The heuristic reported very low computational times compared to those of CPLEX 11.1, with an average of 237 seconds, compared to 7000 seconds respectively.

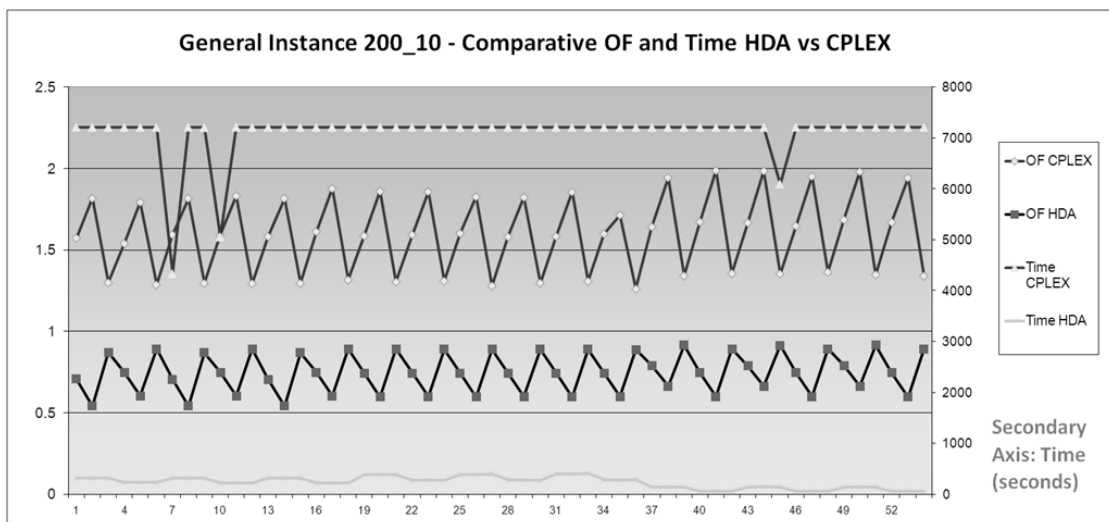


Figure 5. Comparative Chart, General Instances of size 200_10.

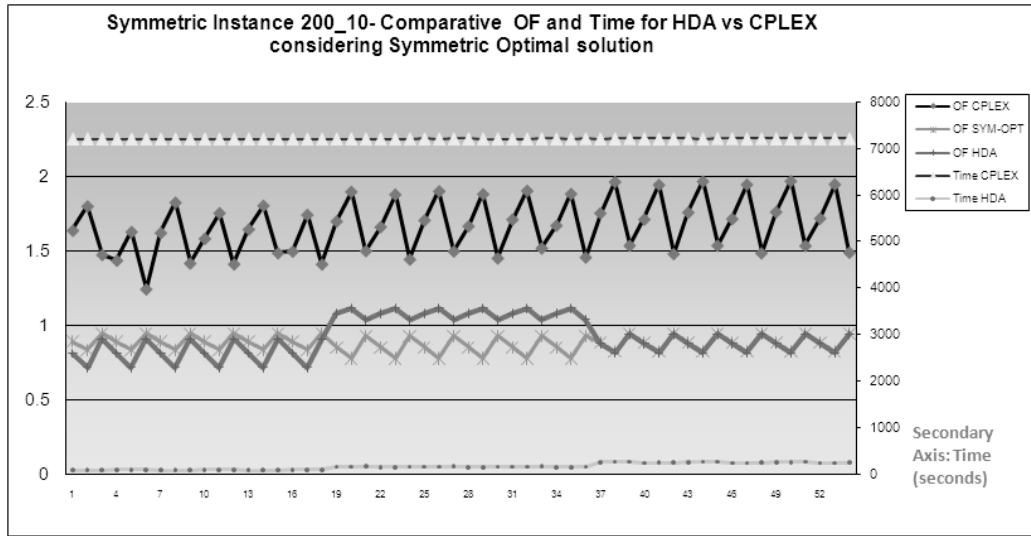


Figure 6. Comparative Chart, Symmetric Instances of size 200_10.

For the Symmetric Instances, the solutions found by CPLEX 11.1 are always worse than those reported by the heuristic, which in most of the instances found the optimal solution or a very nearly optimal solution, as shown in Figure 6. In the figure, the first 18 instances correspond to the first replicate for which the heuristic always found the optimal solution, as was the case for the third replicate. The second replicate corresponds to instances 19 through 36, for which the heuristic found solutions very close to the optimal symmetric solution, as can be seen in Figure 6. Low computational times are reported by the heuristic, with an average of 160 seconds, significantly lower than those for CPLEX 11.1, which always reached the time limit.

For the 450_15 instances, CPLEX 11.1 could not find any feasible integer solution, so we report the linear relaxation (LR) solution reported by CPLEX 11.1. Although the non-integer solution can be used as a lower bound, it is important to recall that a non-integer solution is not comparable to the heuristic solution because it represents a configuration in which fractions of points are allocated to several districts, which in practice does not make any sense for the parcel company. Table 3 and Figure 7 present the results obtained for the general instances. Both the table and the figure report the linear relaxation solution found by CPLEX 11.1.

Size	Metric	URBAN INSTANCES			
		LR-OF1 CPX	OF1 HDA	Time CPX	Time HDA
450_15	Maximum	0.18	0.96	4129.05	3151.53
	Average	0.05	0.82	1898.59	1868.59
	Minimum	0.01	0.59	177.45	1155.56

Table 3. Numerical results- General Instances, 450_15 size.

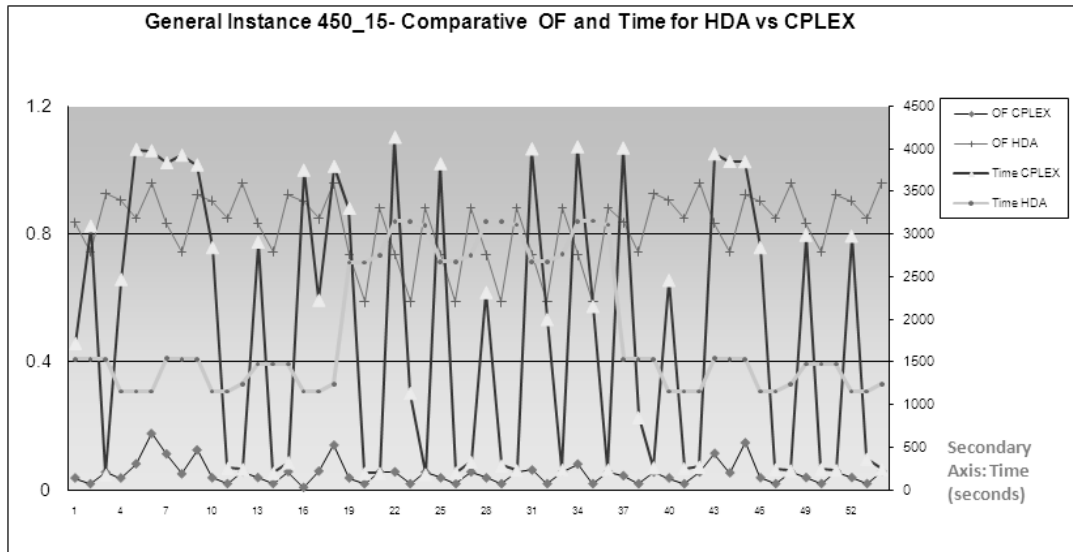


Figure 7. Comparative Chart, General Instances of size 450_15.

We can observe in Table 3 and Figure 7 that the linear relaxation solution found by CPLEX 11.1 is much lower than the solution found by the heuristic and also that for some instances, the computational times are much lower for CPLEX 11.1. However, given that the linear relaxation solution is not comparable and that the lower bound it provides may not be tight, we cannot use these results to make conclusions about the heuristic's performance.

Table 4 and Figure 8 present the results obtained for the symmetric instances. As we can observe, the linear relaxation solution value found by CPLEX 11.1 is much lower than the optimal symmetric solution. This shows that the non-integer solution

found by CPLEX 11.1 does not provide a tight lower bound that can be used for comparison purposes. On the other hand, we can observe that the heuristic found solutions very close to the optimal symmetric solution, with an average gap of 6% and a maximum gap of 21%. We can observe that for this instance size, the computational times for CPLEX 11.1 are lower than those attained by the heuristic. However, this is not a useful comparison because the CPLEX 11.1 solution is not a feasible solution. The heuristic reports reasonable times given the size of the instance, with an average of 1868 seconds and never reaching the time limit. For a strategic level decision problem that does not need to be solved very often, these are fast solution times.

Size	Metric	SYMMETRIC INSTANCES					
		GAP (HDA-OPT)	LR-OF1 CPX	OF1 OPT SYM	OF1 HAD	Time LR-CPX	Time HDA
450_15	Maximum	0.21	0.65	0.95	0.97	1482.77	3151.53
	Average	0.06	0.41	0.84	0.88	393.02	1868.59
	Minimum	0.00	0.19	0.70	0.70	26.35	1155.56

Table 4. Numerical results- General Instances, 450_15 size.

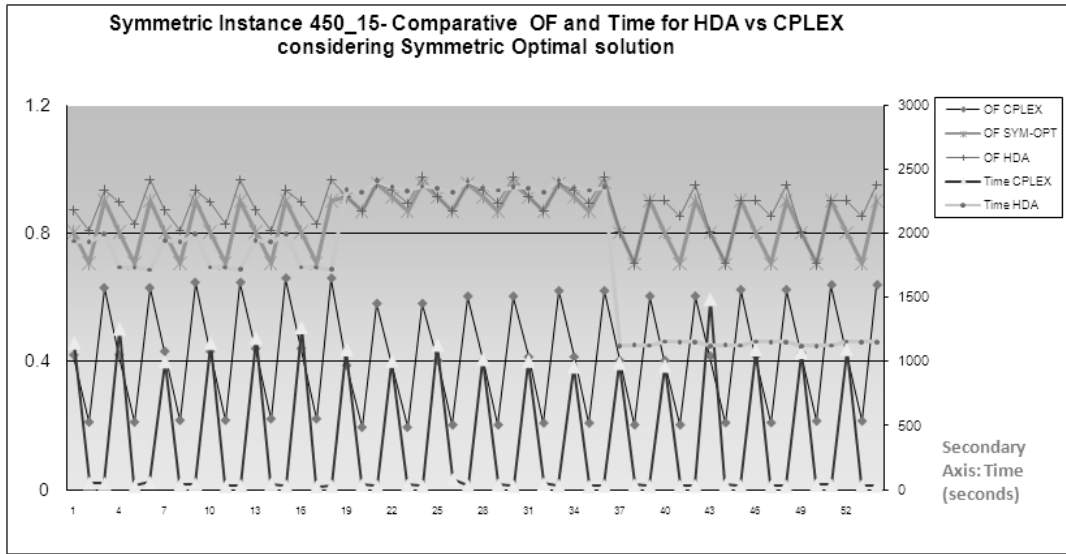


Figure 8. Comparative Chart, Symmetric Instances of size 450_15.

Finally, Table 5 presents a comparison of the results classified by instance size and capacity restrictiveness. For the general instances, we considered only the results that correspond to the 50_5 and 200_10 instances in which CPLEX found a feasible integer solution. For the symmetric instances, we considered the three instance sizes, compared to the optimal symmetric solution.

No large performance difference is observed between both levels of capacity restrictiveness, but in general we can observe smaller gaps for the less

restricted instances, especially for the 200_10 and 450_15 instances, keeping in mind that for the 50_5 instances, both CPLEX and the heuristic found optimal solutions. We must point out that the most difficulty step in finding a solution is the construction of a feasible solution, and the restrictiveness of the capacity limits strongly impacts this part of the procedure. The inclusion of a multi-start procedure increases the probability of being able to find a feasible solution. For all the instances tested, we were always able to find at least one feasible solution.

TYPE OF INSTANCE	SIZE CAPACITY METRIC	50_5						200_10						450_15					
		LESS RESTRICTED			TIGHT			LESS RESTRICTED			TIGHT			LESS RESTRICTED			TIGHT		
		Max	Avg.	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.	Min
SYMMETRIC INSTANCES	GAP CPX -OPT	0.00	0.00	0.00	0.00	0.00	0.00	1.55	0.97	0.55	1.45	0.89	0.37						
	GAP HDA-OPT	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.05	0.00	0.22	0.05	0.00	0.14	0.03	0.00	0.21	0.09	0.02
GENERAL INSTANCES	GAP CPX-HDA	0.00	0.00	0.00	0.00	0.00	0.00	-0.31	-0.51	-0.71	-0.30	-0.51	-0.70						

Table 5. Numerical results- according to capacity tightness.

6. Conclusions and recommendations

We present a mathematical model for a districting problem for a parcel company that operates in the metropolitan area of Monterrey, Mexico. Because the problem is NP-hard, we propose a heuristic procedure to solve the problem. Our computational experiments show that the procedure performs well on the smallest size instances, for which both CPLEX 11.1 and the procedure found an optimal solution. For the medium size instances, the procedure found solutions better than the best integer solutions found by CPLEX 11.1. Low computational times are observed for the heuristic procedure, with a maximum time of less than one hour for the largest size instances tested. We included as part of the experimentation, an instance type in which the optimal solutions correspond to symmetric districts with the same workload content and compactness metric, which helped us to evaluate the performance of the heuristic for the largest size instances, for which CPLEX 11.1 could not find a feasible integer solution, providing only the solution to the linear relaxation.

Some recommendations for further research include modeling the problem as a bi-objective optimization problem and finding the efficient frontier instead of a single solution. Other solution approaches based on decomposition techniques commonly used on large-scale problems may also be implemented and tested. A stochastic version of the problem could also be modeled and solved.

References

- [1] Altman M., "Is Automation the Answer: The Computational Complexity of Automated Redistricting". *Rutgers Computer and Law Technology Journal*, Vol. 23 No.1, 1997, pp. 81-142.
- [2] Feo T.A. and Resende M.G.C. "Greedy randomized adaptive search procedures". *J. Of Global Optimization*, 6: 109-133, 1995.
- [3]Glover, F. (1986) "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research*, 5, 533-549.
- [4] Glover F. and Laguna M. "Tabu Search". Kluwer Academic Publishers, 1997
- [5] Glover F., and Laguna M., "Tabu Search", *Modern Heuristic Techniques for Combinatorial Problems*, C.R. Reeves, ed., Blackwell Scientific Publishing, 71-140, 1993.
- [6]Langevin A, Soumis F. "Design of multiple-vehicle delivery tours satisfying time constraints". *Transportation Research-B*, Vol. 23B No. 2, 1989, pp.123-138.
- [7] Novaes A G.N, Graciolli O.D, "Designing multi-delivery tours in a grid-cell format." *European Journal of Operational Research*, Vol. 119, 1999, pp.613-634.
- [8] Novaes, A.G.N., Souza de Cursi, J.E., Graciolli, O.D. "A continuous approach to the design of physical distribution systems". *Computers & Operations Research*. Vol. 27, 2000, pp. 877-893.
- [9] Muyldermans L, Cattrysse D, Oudheusden D, Lotan T. "Districting for salt spreading operations" *European Journal of the Operational Research*, Vol. 139 No. 3, 2002, pp. 521–532.
- [10] Muyldermans, L., Cattrysse D. and Van Oudheusden, D. "District Design for arc-routing applications". *Journal of Operational Research Society*, Vol. 54, 2003, pp. 1209-1221.
- [11] Haugland, D. Ho, S.C, Laporte G. (2007) "Designing delivery districts for the vehicle routing problem with stochastic demands", *European Journal of Operational Research*, 180(3): 997-1010.
- [12] Moonen M. "Patrol deployment, districting, and dispatching within the urban police: state of the art". Leuven, Belgium: Centre for Industrial Management, Katholieke Universiteit Leuven, 68, HV 8080 .P2 M77, 2004.
- [13] Erkut, E., Neuman S. "Comparison of four models for dispersing facilities". *INFOR*, Vol. 29, No. 2, 1991, pp. 68-86.

Authors' Biographies



Rosa Guadalupe GONZALEZ-RAMIREZ

Rosa G. González is a researcher and professor at the Industrial Engineering School at the Pontificia Universidad Católica de Valparaiso. She holds an undergraduate degree in Industrial Engineering from the Instituto Tecnológico de Morelia, an M.S. in quality and productivity systems from Tecnológico de Monterrey, Campus Toluca, an M.S. in Industrial Engineering from Arizona State University and a Ph.D. in Engineering Sciences from Tecnológico de Monterrey, Campus Monterrey. Her research interests are in the areas of operation research and logistics. She has taught undergraduate level courses in operations research, logistics and port operations.



Neale Ricardo SMITH-CORNEJO

Neale R. Smith is an assistant professor at the Quality and Manufacturing Center at the Tecnológico de Monterey, Campus Monterrey in Mexico. He holds an undergraduate degree in industrial engineering from the University of Arizona and M.S. and Ph.D. degrees in industrial engineering from Georgia Tech. His research interests are in the areas of operations research and logistics. He has taught both undergraduate and graduate level courses in operations research, logistics and supply chain management and statistical process control.



Ronald G. ASKIN

Ronald G. Askin is Professor of Industrial Engineering and Director of the School of Computing, Informatics and Decision Systems Engineering at Arizona State University. He received a BS in IE from Lehigh University, and an MS in OR and a Ph.D. from Georgia Institute of Technology. Dr. Askin is a Fellow of the IIE and former editor of IIE Transactions on Design and Manufacturing. He has authored over 100 publications on the application of operations research and statistical methods to the design and analysis of integrated production control systems. He has received several awards for his textbooks and research, including a NSF Presidential Young Investigator Award and the Shingo Prize for Excellence in Manufacturing Research.



Pablo A. MIRANDA GONZÁLEZ

Pablo Miranda is an assistant profesor at the Industrial Engineering School, Pontificia Universidad Católica de Valparaiso. He holds an undergraduate degree in Industrial Engineering with a major in Transport Engineering, a M.S. and a Ph.D. in Engineering Sciences from the Pontificia Universidad Católica de Chile. His interests are in the areas of operation research, mathematical modeling, supply chain network design and logistics. He has taught undergraduate and graduate level courses in operations research and logistics.



Jose Manuel SÁNCHEZ

Jose Manuel Sánchez is an assistant professor at the Quality and Manufacturing Center at the Tecnológico de Monterrey, Campus Monterrey in Mexico. He has received a BS in Mechanical and Electric Engineering and a MS in Information Systems from Tecnológico de Monterrey, Campus Monterrey, and Ph.D. in Industrial Engineering from Texas A & M University. He is member of the Institute of Industrial Engineers, International Association of Knowledge Engineers and North American Fuzzy Information Processing Society. He is coauthor of books: “Product development design for manufacturing: a collaborative approach to producibility and reliability” and “Handbook of life cycling engineering: tools and technologies”.