Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence

Position Assignment on an Enterprise Level Using Combinatorial Optimization

Leonard Kinnaird-Heether

Chris Dorman

Ford Motor Company

lkinnair,cdorman1@ford.com

Abstract

We developed a tool to solve a problem of position assignment within the IT Ford College Graduate program. This position assignment tool was first developed in 2012 and has been used successfully since then. The tool has since evolved for use with several other position assignment and related tasks with other similar programs in Ford Motor Company. This paper will describe the creation of this tool and how we have applied it, focusing on the need for developing such a tool, and how the continued development of this tool will benefit its users and the company.

Introduction

Many large corporations employ specialized programs designed to transition new employees from college to a professional environment. Many people, particularly undergraduates, leave college with a generalized education in their particular field of study; programs such as these are structured to allow new employees to experience working in different positions, giving them a variety of experiences to guide the early part of their careers. These programs can take the form of internships and also include rotational programs that employees participate in during their first few years of employment, or prior to becoming a full-time employee.

Ford Motor Company offers the Ford College Graduate (FCG) program to recent college graduates. This program allows participants to experience multiple roles over their first few years with the company. The FCG program participants are referred to as FCGs. These programs operate by soliciting supervisors within the organization to offer up potential positions, or rotations, designed to take on a new employee for a fixed length of time, typically one year. The rationale behind rotations is that an employee whom has insight into a variety of business units and the networking contacts gained through those experiences is a better employee in the long term.

The employees that participate in these programs choose positions they are most interested in and are then assigned to one of those positions. One of the more complex issues encountered administering such a program is: how does one assign positions in a manner the employees believe is fair? The main cause of this is that some positions are more desirable than others and are more likely to be selected. Given that many positions will not have enough openings to accommodate all the interested participants, not everyone will be assigned to their most desired position. This brought about a need for a system to assign positions the new graduates believe is egalitarian.

We developed a solution to this position assignment task that uses combinatorial optimization to impartially assign positions based on user preferences. Our paper is organized as follows: first we introduce the problem and why the creation of the position assignment tool was necessary. We then discuss processes that tie-in with the position assignment tool as part of the FCG Carousel process, and discuss how the tool works. We discuss why we chose to use combinatorial optimization and how our problem equates to the classic assignment problem. We review the results of how our tool has performed over time and discuss how our application was developed, deployed and is maintained. Finally we discuss the conclusions of our work, and highlight several areas we feel warrant further study.

Problem Description

The FCG program started at Ford Motor Company in the 1960s and the term FCG was first used in the 1980s. Many business units within Ford offer this program and fine-tune it to their particular business needs. The information technology business unit program known as the IT FCG program has embraced the idea that employees with a diverse knowledge of the business are more valuable employees and have adjusted the methodology to assign positions over time. At some point the idea of empowering the participants to choose their own roles was introduced. Early on these choices were rudimentary. The program did not possess the structure or the population to require any complex means to perform the position assignment task; the administrators of the program hand-assigned positions to the FCGs. This solution worked well when the number of participants was small, however as the program grew eventually the number of possible participant to position combinations made assigning positions by hand no longer a feasible option.

The program administrators decided to create a software tool to automate the process and followed an open ideal by including the program participants in the development. Early attempts were made using an off-the-shelf software solution, but it soon proved to be insufficient for processing large numbers of position and participant combinations. It became necessary to develop a custom solution to resolve the problem, providing genesis to the position assignment tool that is currently in use.

The decision to move from a manual solution to a computational one was primarily a matter of practicality. It became impractically time-consuming for humans to process the position assignments manually once there were more than a handful of participants. Using an artificial intelligence based approach with a combinatorial optimization algorithm provides a solution that is more than capable of managing all foreseeable future needs. In addition, by creating a modular system in which the AI acts only as a solution generator leaves open the possibility of using approaches other than combinatorial optimization in the future.

Application Description

The position assignment tool that we developed provides a solution to the problem of equitably matching participants with positions that they have interest. The tool was initially designed for use with the IT FCG program and their annual position assignment task. The tool and the related processes are maintained and operated by a subset of current members of the IT FCG program, known as the IT FCG Carousel group. The position assignment process, referred to as the FCG Carousel process, is shown in Figure 1.

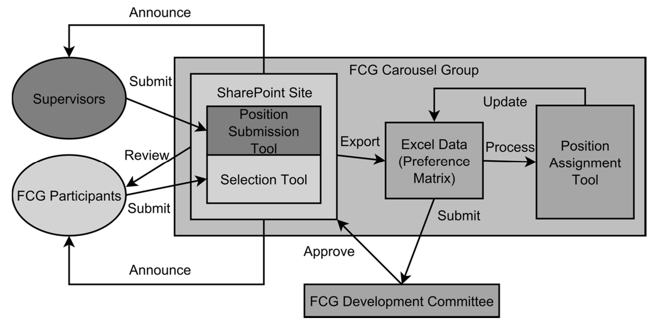

Figure 1

Each year, FCGs are given the opportunity to select options for their upcoming year. This year of work is referred to as a rotation. Each position is given a designation that roughly defines what area of work that the position entails. These designations are *Technical*, *Very Technical*, *Business/Process*, and *Very Business/Process*. Essentially the technical roles are, as one would surmise, more technologically focused, such as writing application source code, configuring servers and so forth. Whereas the business roles focus more on organization and maintaining business processes, for example writing security documents and defining workflows. The FCGs are required to follow certain guidelines when selecting their new roles, and these position classifications play a part. The guidelines related to these classifications take effect based on where the FCG is in their tenure in the program, and are designed to induce participants to explore new areas, thus giving them a broader understanding of the business. The diagram in Table 1 shows these guidelines.

FCG Year	Position Selection Guidelines
First	No restriction.
Second	If current position is technical, must choose business process, and vice-versa.
Third	Must choose position not already held.

Table 1: FCG Position Selection Guidelines by Year.

The supervisors and participant FCGs both use SharePoint based online tools to input the requested data. The process typically begins in early January with an email sent to supervisors soliciting positions for the upcoming Carousel process. The supervisors are typically given approximately a month to provide a written description of the position that includes an inventory of skills desired, and the benefits they feel the participant will get out of this rotation. The skills listed by the supervisor are for the participant's reference to assist them in choosing a rotation they can be successful in. However no mechanism exists to safeguard that the participant has experience in a particular skill.

Once all the rotation information is collected from the supervisors, it is disseminated to the FCG population via websites and documents located on the corporate intraweb. The FCGs are then given approximately one month to review the positions, meet with prospective supervisors, and enter their selections using a SharePoint based selection tool. This tool is designed with logic to enforce the guidelines referred to previously, such that the FCG will be unable to view or select positions they are restricted from by the guidelines.

User Position Preference Data

Preference data in its raw form consists of participants rankings of positions. Rather than have participants rank all the positions, the system only requires that participants rank their top *n* choices. The value of *n* is variable and can be increased or decreased at will, but the most common value is 4.

Weight Function

Once the preference data has been collected, the position assignment tool then must calculate the preference scores using the weight function shown in Figure 2. This particular weight function was chosen because it closely approximates the weight table that was used in previous versions of the position assignment tool, shown in Table 2. In this example, the participant's most desired choice is represented by the position rank of 4, and their fourth top choice by a position rank of 1. Participants may not rank the same position more than once. The seniority value is representative of a participant's tenure in the program. Participants just starting the program have a seniority value of 1, and participants in their final year have a seniority value of 3.

$$\boldsymbol{W} = \frac{\sqrt{\boldsymbol{R}} * (\boldsymbol{3S} + \boldsymbol{3})}{\boldsymbol{2}} - \boldsymbol{2}$$

Where:
R: Position Ranking $\in$ *{1,2,3,4}*
S: Seniority Level $\in$ *{1,2,3}*
W: Weight Value

Figure 2

Seniority	Position Rank			
	4	*3*	*2*	*1*
1	4	3	2	1
2	8	7	6	5
3	12	11	10	9

Table 2: Weight table approximated by the weight function.

Weight Matrix

After calculating each participant's preference weight value, the data are stored in a weight matrix, such as the example shown in Table 3.

Participant	Position					
	1	*2*	*3*	*4*	*5*	*6*
A	12	9	0	11	0	10
B	8	0	5	0	6	7
C	7	8	0	0	6	5
D	0	1	3	2	4	0

Table 3: Example weight matrix.

Using this weight matrix, the position assignment will find the best one-to-one match of participants to positions. Each potential solution is scored by the sum of the weights of each assignment, with the intent of finding the potential solution with the maximum weight. In the example shown in Table 4, the potential weight is 31 (the sum of the highlighted cells). We call this figure the potential satisfaction score, and compare it to the actual satisfaction score after the tool is complete as a means to measure the success of a particular position assignment task.

Participant	Position					
	1	*2*	*3*	*4*	*5*	*6*
A	**12**	9	0	11	0	10
B	8	0	5	0	6	**7**
C	7	**8**	0	0	6	5
D	0	1	3	2	**4**	0

Table 4: Example weight matrix with example position assignment highlighted.

Once all the data regarding the positions and preferences is collected, it is exported in the form of a preference matrix, similar to the weight matrix of table 4, but without the weight function applied. The data are exported as an Excel spreadsheet for ease of use. The tool uses the weight function with the individual preferences to create the weight matrix and then performs a combinatorial optimization using the Hungarian (or Kuhn-Munkres) algorithm (Kuhn 2005). The tool requires that the number of positions available be greater than the number of participants to function, and expects, but does not require, that all participants will possess valid preference data. The tool is also constrained such that in a valid potential solution, each participant must be assigned to a position, and each position must have either one or zero participants assigned to it.

The final matching is written out to another sheet in the Excel file, which also contains formulae designed to calculate the total satisfaction rating of the optimum solution, as well as other metrics related to what percentage of FCGs received their first choice, second choice, etc. The satisfaction rating in this case is the ratio of the satisfaction score of a position assignment over the satisfaction score of a perfect assignment for the same assignment problem, where all participants receive their top choice. In practice, such a perfect assignment is very unlikely to occur, but the fact that it could occur makes it useful for benchmarking how well the system performs the assignment task. This spreadsheet and the associated metrics values are presented to the FCG Development Committee for final approval. This committee retains the option of altering the final results, but has not done so since the adoption of the automated position assignment system. The approved final results are released to the supervisors and FCGs in early May. The FCGs have several months to prepare for their new positions starting in early June for new participants and August for existing participants.

Use of AI Technology

The problem that the position assignment tool is designed to solve is similar to the classic assignment problem (Burkard, Dell'amico and Martello, 2012). The primary difference is that the classic assignment problem is defined as a minimization problem. However, all of our ancillary processes related to the position assignment task, particularly the weight function, treat this problem as a one of maximization. The choice of this is purely aesthetic, as it is more appealing to the end users to think of this task as finding the best or maximum possible value. Fortunately it is relatively easy to convert this to a minimization problem by creating a new weight matrix, derived from subtracting the results of the original weight matrix from the maximum value of the original weight matrix. Table 5 shows an example, derived from the previous example.

Participant	Position					
	1	*2*	*3*	*4*	*5*	*6*
A	**0**	3	12	1	12	2
B	4	12	7	12	6	**5**
C	5	**4**	12	12	6	7
D	12	11	9	10	**8**	12

Table 5: Example position assignment weight matrix converted to a minimization problem

Since combinatorial optimization techniques are already used to solve a variety of real world problems, including course allocation, resource allocation, vehicle scheduling, and task scheduling, this motivated the decision to use it as the solution generator for the position assignment tool (Hoshino and Raible-Clark, 2014) (Cambazard, O'Sullivan and Simonis, 2013) (El Hachemi, Gendreau and Rousseau, 2011) (Mencía, Sierra and Varela, 2013.)

The previous solution was based on Frontline's Premium Solver Platform. This solution proved insufficient due to the fact that it imposes an artificial limit of 8000 on the number of unknown decision variables it can resolve. In our case the number of decision variables is equivalent to the number of positions times the number of participants, We had 80 participants in 2012 and 196 positions, giving us a total of 15680 possible decision variables, thus greatly exceeding the artificial limit imposed by Frontline.

We chose the Hungarian method as the functional heart of the position assignment tool due to its ability to resolve our functional constraints. Primarily, this method is guaranteed, by its own definition, to find an optimal solution, should one exist. Moreover this algorithm is capable of handling the increased complexity of the position assignment task, which is brought on by the increase in population and available positions.

Depending on the implementation of the algorithm, this characteristic can introduce some bias into the solution. Because a given set of data may have more than one optimal solution, the implementation may bias itself toward a particular solution. Multiple optimal solutions occur when the resulting potential satisfaction rating of each optimal solution is equal, but the actual position assignments are different. Depending on the implementation of the Hungarian algorithm, an order bias can be introduced here. In the case where the system determines that the potential gain of placing individual A or individual B in a given position is of equal value the classic implementation of the Hungarian algorithm will bias toward individual A, simply because A comes first in the ordering of individuals in the weight matrix. To avoid this bias we randomize the order of individuals prior to generating the position assignment.

The other qualification that led to the choice the Hungarian method was its ability to handle the position and participant constraints. As mentioned previously, any valid solution must maintain the constraints that every participant must be assigned one position, and every position must be assigned one or zero participants, something which is possible to achieve in the implementation of the Hungarian algorithm. In the case where a position may have more than one available slot, we create multiple columns in the weight matrix for the same position, thus maintaining the constraint on position assignment, without adding any major computational complexity.

Application Use and Payoff

This version of the position assignment tool was developed in 2012 and has been in active use since then. For the IT FCG program, the tool is used once yearly as part of the FCG Carousel process. The metrics related to total satisfaction are used as a bellwether of how well the process is performing. Prior to 2011 performance metrics were either not kept or were largely unavailable. Anecdotally it has been said that participants were usually satisfied with the assignment results, and it was primarily the time cost of doing the assignment task manually that led to the change-over from the manual method to the automated Frontline Solver-based method. In Table 6, the metrics for the years 2011-2014 are shown.

Year	Pop.	Satisfaction Rating	Choice			
			First	*Second*	*Third*	*Fourth*
2011	59	80.0%	55.9%	25.4%	16.9%	1.7%
2012	80	82.2%	63.8%	23.8%	8.8%	3.8%
2013	120	81.6%	65.0%	22.5%	10.8%	1.7%
2014	145	77.2%	55.9%	28.3%	9.7%	6.2%

Table 6: 2011-2014 FCG Carousel process statistics.

This data shows that since its implementation in 2012, the new position assignment tool has been able to complete the assignment successfully and it has more or less maintained the level of satisfaction experienced with the previous version of the tool.

The position assignment tool has also been modified for use with position assignment tasks in other areas. These include the IT Summer Intern program, as well as the Ford Finance College Foundational Program (CFP.) Performing the position assignment for the intern program is identical to the FCG task, with the exception that interns do not possess seniority, meaning a different weight function is used. Table 7 shows the results of the Summer Intern position assignment from 2012-2014, performed using our new position assignment tool.

Year	Population	Satisfaction Rating
2012	25	59.7%
2013	48	74.0%
2014	54	51.7%

Table 7: 2012-2014 IT Summer Intern Program process statistics.

These satisfaction ratings are much lower than what we have seen with the FCG program. The reasons for these lower ratings is not due to a failure of the position assignment tool, but rather an issue of the program itself relating to the viability of the position offerings. This issue will be further discussed at length in the *Future Work* section of this paper.

The CFP task is slightly different in that it is already formed as a minimization problem. The CFP program uses very large numbers in the weight matrix as a means of preventing participants from ranking positions they are ineligible for by program rules. This enforces a constraint on the participants that is not necessary in the FCG solve since this task is already handled by the online selection tool. The choice of using very large numbers means that any solution that contains an invalid choice will invariably have a higher weight matrix score than any solution containing all valid choices. Since this is a minimization task, this essentially prevents all solutions with invalid choices from being considered. This allows the CFP program to induce participants to explore other areas, in much the same way as the IT FCG program uses the business and technical role designations. Table 8 shows the results of the CFP position assignment process from 2012-2014.

Date	Pop.	Score	Choice					Assigned to Non-Choice
			First	*Second*	*Third*	*Fourth*	*Fifth*	
2012-Feb.	31	442	51.6%	22.6%	12.9%	-	-	12.9%
2013-Mar.	20	223	65.0%	25.0%	0.0%	-	-	10.0%
2013-Jun.	41	30	41.5%	41.5%	14.6%	2.4%	0.0%	0.0%
2014-Feb.	15	29	40.0%	40.0%	6.7%	13.3%	-	0.0%
2014-Sept.	14	128	50.0%	7.1%	14.3%	14.3%	7.1%	7.1%

Table 8: 2012-2014 CFP Program process statistics.

The score column in Table 8 relates to the sum of the weights of the position assignment that was used for the corresponding year. This value is similar to the satisfaction score in the IT FCG position assignment process, but since the assignment task is one of minimization, a lower score represents a better outcome. One unique point about the CFP program assignment task is that the number of choices is variable. The CFP program allows participants to choose from position areas, rather than specific positions, and each position area has a fixed number of available positions that must be filled. The result is under some circumstances a participant will be assigned to a position area that they did not choose. To alleviate the problem the CFP program organizers offered more choices; this adjustment has improved the results but has not solved the issue.

The position assignment tool has also been modified to perform a *buddy* assignment task for both the IT FCG program and the IT Summer Intern program. This ancillary task is similar to the position assignment task, but rather than attempting to assign participants to their desired positions, this task attempts to pair each participant with a buddy, which is essentially an employee who volunteers to serve as a friendly point of contact to the new employee or intern. This matching is performed by comparing various attributes of each employee. These attributes include sharing the same supervisor, working group, office building, and university affiliation. Based on these scores the system attempts to find the best match between buddies and recipients. There are currently no metrics to determine the fitness of this process, other than anecdotal evidence that people find it easier to communicate when the person they are paired with is collocated with them.

Application Development, Deployment and Maintenance

The initial development was done in early 2012 by a single developer, creating a standalone Java application. This task took approximately 20 hours to complete. An additional MATLAB based tool was concurrently developed, but later abandoned because of lack of interest and knowledge of MATLAB programming. The tool has undergone other small improvements by a team of developers in the meantime, mostly focusing on integrating the program in with the other parts required for the FCG Carousel process. The tool is currently run as a standalone Java application that can be run in either a desktop or server based environment. Since it processes data to and from and Excel spreadsheet, it can be run separately from the other support parts of the system, provided the required preference and position data are present.

The position assignment tool so far has required minimal maintenance. The exception being minor formatting changes to coincide with changes in the Carousel process, or for integrating the tool with other process, such as CFP assignment and buddy assignment. The tool was designed as a standalone process that can be updating with a minimum of impact when the other aspects of the overall process are tweaked from time to time, as one would expect in a vibrant program. Updates are provided by a group of FCGs who volunteer to maintain and support the Carousel process during their tenure.

Conclusions and Further Work

The objectives of this paper are two-fold; to explain how the use of combinatorial optimization can be used to provide a best-fit position assignment on an enterprise-level, and to demonstrate its benefits as well as issues that can arise from its use. The position assignment tool we developed allows the IT FCG program to continue to use their Carousel process unchanged, whereas it could not have using the previous version position assignment software. The tool has also shown its ability to adapt to other related tasks with reasonable levels of success. However, there are some issues with the process, unrelated to the position assignment tool.

The falling satisfaction rating scores observed in the IT FCG program (Table 6) and more so in the IT Summer Intern program (Table 7) are likely related to the increasing membership of both programs, and a decrease in position availability. Like many instances where people are asked to choose their preferences from a shared set of resources, collisions within preferences is likely to occur. In this case we assume that since each user is attempting to maximize their career satisfaction there will be certain positions perceived to be better than others. These positions receive more attention than others, leading to conflict over key positions. Anecdotally, we have noticed over the past few years that certain positions, typically ones associated with popular technological trends, tend to garner more interest than others. Since these positions often have only one opening, if several people choose that position only one FCG is receiving their top choice and the others are receiving a less desirable choice, which we can see represented in the satisfaction scores.

One possible solution we attempted to resolve this was to provide a wide range of positions. We felt that when soliciting supervisors for positions that more would be better, and by offering a wider array of positions, we hoped to diffuse the trend of participants gathering on specific positions. It was decided that a 2-to-1, position to participant ratio would be a good guideline for the number of positions. However, this is easier said than done, and it has since proven difficult to find enough meaningful positions to meet this goal.

The Table 9 shows a breakdown of position to participant ratios, along with the corresponding satisfaction ratings for the years 2011-2014.

Year	Positions Available	Number of Participants	Ratio	Satisfaction Rating
2011	100	59	1.695	80.0%
2012	196	80	2.45	82.2%
2013	197	120	1.642	81.6%
2014	271	145	1.869	77.2%

Table 9: Comparison of Position-Participant Ratios and Satisfaction Ratings.

This data shows that when the position to participant ratio was less than 2, satisfaction ratings for that year were slightly lower, but it falls short of showing a direct correlation between this ratio and the satisfaction ratings.

Investigating further, we noticed that many positions receive little or no interest at all. In Table 10 we show the number of positions that were selected by less than 2% of users and those that were not selected at all.

Year	Total Positions	Pop.	<2% Interest	0% Interest	Effective Positions	Effective Ratio	Satisfaction Rating
2011	100	59	48	3	49	0.831	80.0%
2012	196	80	46	66	84	1.05	82.2%
2013	197	120	127	15	55	0.458	81.6%
2014	271	145	161	62	48	0.331	77.2%

Table 10: Comparison of Effective Position-Participant Ratios and Satisfaction Ratings.

When removing the less popular positions from the total count, we get an effective positions count, and its subsequent ratio to participants. This ratio falls short of having a direct correlation to the satisfaction rating, but definitely provides a better indicator than the simple ratio.

In the future we would like to study how user selections are affected by the quality of position offerings. In particular we could investigate work done on user equilibrium (Wardrop 1968) (Cominetti, Facchinei, Lasserre, 2012.) to see if it has any application with our work. User equilibrium is more often used in traffic management as mainly deals with methods of finding a point of equilibrium between users attempting to use similar routes on a system of roads. Finding an adequate means of sharing limited resources while keeping the participants content, is very much in line with the issues we are facing.

A question that often arises from this work is whether or not participants are able to effectively game, or influence the outcome of the system, and if so, how? The ways that the system could be influenced fall into two general categories. One relates to a fault with the technical process itself, and another relates more to social engineering. From a technical perspective, there is no evidence that individuals are working to influence the final outcome, but it does not mean that it is not happening. It is very likely that this system is susceptible to collusion between individuals, and that is an area of interest for future research. From the social engineering perspective, there is already anecdotal evidence of individuals attempting to influence the outcome of the position assignment in ways that are totally above board. One method has an individual ranking all of their position choices in a particular popular working group, thus guaranteeing themselves a spot somewhere in that group. Once in that group the individual works to alter their job duties from the original specifications, such that their job description effectively becomes the position they truly desired in the first place. There is also evidence of individuals with more esoteric taste in positions soliciting supervisors to submit positions to the Carousel, primarily because that individual feels that no-one else would be interested in it, thus leaving that position completely open for them to take. Individuals are often known to take positions from a less desired category (business or technical) first, thus giving them a higher weight when competing for positions that they truly desire in subsequent years.

Another area of interest centers on the performance of position assignment tool. We have not confirmed it is robust enough to handle an increased load. The FCG program continues to grow and at some point it may become so large that the performance of this tool may suffer. In its current state the position assignment tool runs as a single-threaded process, primarily because the implementation of the Hungarian algorithm it relies on was also written as single-threaded. It might be necessary in the future to investigate other similar algorithms that support parallel processing (Kollias, et. al., 2014.)

The FCGs give generally positive feedback about the Carousel process; however, large corporations are under continued pressure to improve their employee development programs to recruit the best employees. We could explore other techniques for solving the assignment problem using genetic algorithms, cultural algorithms, or other techniques that may lead to better results. (Bowman, Briand, and Labiche, 2010) (Soza, et. al., 2011) (Reynolds and Kinnaird-Heether, 2013.)

References

Kuhn, H. W. 2005. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, *52*(1), 7-21.

Burkard, R., Dell'Amico M. and Martello S. 2012. *Assignment Problems* (Revised reprint). Philadelphia, PA.: SIAM.

Hoshino, R., & Raible-Clark, C. 2014. The Quest Draft: an Automated Course Allocation Algorithm. In *Twenty-Sixth IAAI Conference*.

Cambazard, H., O'Sullivan, B., & Simonis, H. 2013. The Deployment of a Constraint-Based Dental School Timetabling System. In *Twenty-Fifth IAAI Conference*.

El Hachemi, N., Gendreau, M., & Rousseau, L. M. 2011. A hybrid constraint programming approach to the log-truck scheduling problem. *Annals of Operations Research*, *184*(1), 163-178.

Mencía, C., Sierra, M. R., & Varela, R. 2013. An efficient hybrid search algorithm for job shop scheduling with operators. *International Journal of Production Research*, *51*(17), 5221-5237.

Wardrop, J. G. 1968. Journey Speed and Flow in Central Urban Areas. *Traffic Engineering and Control*. Vol. 9, No. 11.

Cominetti, R., Facchinei, F., & Lasserre, J. B. 2012. Wardrop and Stochastic User Equilibrium. In *Modern Optimization Modelling Techniques* 213-220. Springer Basel.

Kollias, G., Sathe, M., Schenk, O., & Grama, A. 2014. Fast parallel algorithms for graph similarity and matching. *Journal of Parallel and Distributed Computing*, *74*(5), 2400-2410.

Bowman, M., Briand, L. C., & Labiche, Y. 2010. Solving the class responsibility assignment problem in object-oriented analysis with multi-objective genetic algorithms. *Software Engineering, IEEE Transactions on*, *36*(6), 817-837.

Soza, C., Becerra, R. L., Riff, M. C., & Coello Coello, C. A. 2011. Solving timetabling problems using a cultural algorithm. *Applied Soft Computing*, *11*(1), 337-344.

Reynolds, R. G. & Kinnaird-Heether, L., 2013. Optimization Problem Solving with Auctions in Cultural Algorithms. *Memetic Computing*, 5(2), 83-94.