A Firefly Algorithm to Solve the Manufacturing Cell Design Problem

Ricardo Soto[1,2,3], Broderick Crawford[1,4,5], Jacqueline Lama[1], and Fernando Paredes[6]

[1] Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
[2] Universidad Autónoma de Chile, Santiago, Chile
[3] Universidad Cientifica del Sur, Lima, Peru
[4] Universidad Central de Chile, Santiago, Chile
[5] Universidad San Sebastián, Santiago, Chile
[6] Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
{ricardo.soto,broderick.crawford}@ucv.cl
jacqueline.lama.g@mail.pucv.cl
fernando.paredes@udp.cl

Abstract. The Manufacturing Cell Design Problem (MCDP) consists in creating an optimal design of production plants, through the creation of cells grouping machines that process parts of a given product. The goal is to reduce costs and increase productivity by minimizing movements and exchange of material between these cells. In this paper, we present a Firefly Algorithm (FA) to tackle this problem. The FA is a recent bio-inspired metaheuristic based on the mating behavior of fireflies that employ its flashing capabilities to communicate with each other or attract potential prey. We incorporate efficient transfer and discretization methods in order to suitable handle the binary domains of the problem. Interesting experimental results are illustrated where several global optimums are reached for a set of 90 well-known MCDP instances.

Keywords: Manufacturing Cell Design, Firefly Algorithm, Metaheuristics, Optimization.

1 Introduction

Group Technology refers to the grouping of parts or products into families, which are processed in a miniature factory called cell [19]. In order to increase production efficiency, the underlying identity of components are exploited; such as shapes, dimensions, routes of processes, etc. The awareness that many problems can be similar and grouped together allows for the search of a solution to satisfy a set of problems in the same time; achieving time and effort optimization. In this context, the Manufacturing Cell Design Problem (MCDP) involves the creation of an optimal production plant design, through the organization of machines that process parts of a given product in production cells. The goal is to reduce costs and increase productivity by minimizing movements and exchange of material between those cells.

This paper focuses on solving the MCDP by using the Firefly Algorithm (FA), which is a recent swarm-based metaheuristic inspired on the simulation of characteristic behavior of the fireflies. Each firefly represents a possible solution to the problem, which are randomly generated. Through the movement behavior, the fireflies move towards the one they feel most attracted for, which allows to update their current solution with a better one. Interesting experimental results are illustrated where several global optimums are reached for a set of 90 well-known MCDP instances.

This paper is organized as follows: In Section 2, we present the related work followed by the mathematical formulation of the MCDP. Section 4 introduces the FA and their basic behaviors. Finally, we present experimental results, conclusions and future work.

2 Related Work

The cell formation problem has been subject of considerable research, where the production flow analysis proposed by Burbidge's in 1963 [6], becomes one of the first procedures to solve this problem. His method uses the machine-part incidence matrix, and it is reorganized in a Block Diagonal Form (BDF) [22]. Analogous approaches try to identify groups of machines, most of them are based on the machine-part incidence matrix. Various examples can be seen in this context by using mathematical programming [1,3,4,14,15] and goal programming [16,17]. Different metaheuristics have also been reported to solve different instances of the MCDP, e.g. tabu search [2,13], particle swarm optimization [10], and genetic algorithms (GA) [20]. Some hybridizations can also be found such as GA with a branch and bound algorithm [5], local search and GA [12], and simulated annealing with GA [21]. Finally, some approaches based on constraint programming and SAT have also been reported [18].

3 Manufacturing Cell Design Problem

The Manufacturing Cell Design Problem (MCDP) involves processing a collection of similar parts on a dedicated group of machines or manufacturing processes. A manufacturing cell can be defined as an independent group of functionally dissimilar machines, located together on the floor, dedicated to the manufacture of a family of similar parts. Furthermore, a part family can be defined as a collection of parts which are similar either because of geometric shape and size or because similar processing steps are required to manufacture them [11].

3.1 Problem Statement

The goal of the MCDP is to minimize movements and exchange of material between cells, in order to reduce production costs and increase productivity. The idea is to represent the requirements of machine parts processing through a matrix called machine-part. The main goal of this matrix is the grouping of machines for forming sets of machines and workpieces, so the number of transport of parts through the cells is minimized. This reorganization is intended to minimize the total number of movements between cells and the variation of load inside of them, which results in the formulation of two new matrices called machine-cell and part-cell. A rigorous mathematical formulation of the problem of grouping machine-part is given by the optimization model depicted in the following [18]. Let:

- M, be the number of machines.
- P, be the number of parts.
- C, be the number of cells.
- i, be the index of machines $(i = 1, 2, ..., M)$.
- j, be the index of parts $(i = 1, 2, ..., P)$.
- k, be the index of cells $(i = 1, 2, ..., C)$.
- M_{max}, be the maximum number of machines per cell.
- $A = [a_{ij}]$, be the binary machine-part incidence matrix, where:

$$a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ process the part } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

- $B = [b_{ik}]$, be the binary machine-cell incidence matrix, where:

$$b_{ik} = \begin{cases} 1 & \text{if machine } i \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- $C = [c_{jk}]$, be the binary part-cell incidence matrix, where:

$$c_{jk} = \begin{cases} 1 & \text{if part } j \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The objective function models the minimization of the part movements among cells as depicted in Eq. 4.

$$min \sum_{k=1}^{C} \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij} c_{jk} (1 - b_{ik}) \tag{4}$$

This objective function is subjected to three constraints as depicted in the following, where Eq. 5 states that each machine belongs to one and only one cell. Eq. 6 guarantee that each part is assigned to one and only one cell, and Eq. 7 determines the maximum number of machines that a cell could has.

$$\sum_{k=1}^{C} b_{ik} = 1, \forall i \tag{5}$$

$$\sum_{k=1}^{C} c_{jk} = 1, \forall j \tag{6}$$

$$\sum_{i=1}^{M} b_{ik} \leq M_{max}, \forall k \tag{7}$$

4 Firefly Algorithm

The Firefly Algorithm (FA), introduced in [23], is a bio-inspired metaheuristic based on the mating or flashing behavior of fireflies. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The flashing light is produced by a process of bioluminescence, and the true functions of such signaling systems are still debating. However, two fundamental functions of such flashes are to attract mating partners (communication) and to attract potential prey.

By idealizing some of the flashing characteristics of fireflies, firefly-inspired algorithm use the following three idealized rules [24]:

i. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
ii. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
iii. The brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness of each firefly is proportional to the value of the objective function. In case of minimization problem, brightness of each firefly is inversely proportional to the value of the objective function.

4.1 Attractiveness

In the FA, the main form of attraction is described by a decreasing function, which is proportional to the *light intensity* seen by adjacent fireflies. This is expressed in the following general form [7]:

$$\beta(r) = \beta_0 \exp[-\gamma r^2] \tag{8}$$

Where β_0 is the attractiveness at $r = 0$ and γ is a absorption coefficient, which controls the decrease of the *light intensity*.

4.2 Distance

The distance between any two fireflies p and q at positions x_p and x_q respectively, can be defined as a Cartesian distance as follows [7]:

$$r_{pq} = \sqrt{\sum_{s=1}^{d} (x_p^s - x_q^s)^2} \tag{9}$$

Where x_p^s is the sth component of the spatial coordinate of the pth firefly and d is the is the number of dimensions.

4.3 Movement

The movement of a firefly p, when attracted to another more attractive (brighter) firefly q, is determined by [7]:

$$x_p^{t+1} = x_p^t + \beta(r)(x_q^t - x_p^t) + \alpha(rand - \frac{1}{2}) \tag{10}$$

Where x_p^{t+1} is the firefly position of the next generation. The first term in the equation is the current position of a firefly x_p, the second term denotes a firefly's attractiveness and the last term is used for the random movement if there are not any brighter firefly. The randomness parameter is represented by α and $rand$ is a random number generated uniformly distributed between 0 and 1.

4.4 Binarization

When the firefly p moves toward firefly q, the position in that dimension of the firefly p is changed from a binary number to a real number. Therefore, the real number will be altered by the following transfer function, which limits the value of this position between 0 and 1 [9]:

$$T(x_p^s) = |\tanh(x_p^s)| \tag{11}$$

Then, the position of the firefly p in the sth dimension is updated using the following discretization method:

$$x_{new}^s = \begin{cases} 1 & \text{if } rand \leq T(x_p^s) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

4.5 Binary Firefly Algorithm

Based on the three rules that idealize the natural behavior of fireflies, the basic steps for FA can be summarized as the pseudo-code shown in Algorithm 1.

Algorithm 1 Binary Firefly Algorithm

Initialize algorithm's parameters:
– Number of fireflies (n),
– Maximum number of generations ($MaxGen$),
– $\beta_0,\ \gamma,\ \alpha$.
Generate initial population of fireflies $x_i,\ (i = 1, 2, ..., n)$.
Light intensity of firefly I_i at x_i is determined by value of *objective function* in Equation (4).
while ($t < MaxGen$) **do**
  for ($p = 1 : n$) **do**
    for ($q = p + 1 : n$) **do**
      if ($I_q > I_p$) **then**
        Move firefly i towards firefly j according to Equation (10).
        Obtain attractiveness in Equation (8), which varies with distance r according to Equation (9).
        The obtained values are binarized by Equation (11) and (12).
      end if
      Evaluate new solutions and update *light intensity*.
    end for
  end for
  Rank the fireflies and find the *current best value*.
end while
Post-process results and visualization.

5 Experimental Results

The FA, as well as the MCDP, was encoded in Java and executed in a 2.40 GHz Intel Core i7 3630QM processor with 12 GB RAM machine running Windows 8.1. The algorithm performance was evaluated in an experimental way, following the execution of 90 instances of the MCDP taken from [4] (10 problems using different M_{max} and C values). Parameter setting for the implemented FA is based on the work done on [8] and [24], which is the following: $\beta_0 = 1$; $\gamma = 1$; $\alpha = 0.2$; $n = 25$; y $MaxGen = 50$. Values obtained after the experimental phase are summarized in Tables 1 and 2, where 'O' denotes the global optimum given in [4], 'F' the best value obtained by the proposed FA, 'A' the average of obtained optimums, and 'RPD' the Relative Percentage Deviation, which is computed as follows:

$$RDP = \frac{(Z - Z_{opt})}{Z_{opt}} \times 100$$

where Z_{opt} is the best known optimum value and Z is the best optimum value reached by FA.

The results exhibit that the proposed approach is able to reach the global optimum for all the 90 tested instances. Analysis of the 'A' column in both tables reveal that only 11 of 90 problems obtained results that differ from the global optimum, however, such a difference turns out to be minimal. Fig. 1 shows the behavior of FA when seeking the *current best value* for problem 1, whose parameters are: $M_{max} = 9$ and C = 2. Thanks to the FA's operating mode, a rapid convergence to the optimal value is obtained, because the *current best value* is minimized in different fireflies in a same generation. In contrast, when working with C = 3, the *current best value* decreases less abruptly, which can be seen in Fig. 2, whose parameters for problem 6 are: $M_{max} = 7$ y C = 3. Despite of differences when dealing with C = 2 or C = 3, the optimum is reached in most cases before 50 generations, demonstrating the efficiency of the proposed approach.

Table 1: Experimental Results I.

	C = 2																			
P	$M_{max} = 8$				$M_{max} = 9$				$M_{max} = 10$				$M_{max} = 11$				$M_{max} = 12$			
	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)
1	11	11	11.4	0.00	11	11	11	0.00	11	11	11	0.00	11	11	11	0.00	11	11	11	0.00
2	7	7	7.6	0.00	6	6	6	0.00	4	4	4	0.00	3	3	3	0.00	3	3	3	0.00
3	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00	3	3	3	0.00	1	1	1	0.00
4	14	14	14	0.00	13	13	13	0.00	13	13	13	0.00	13	13	13	0.00	13	13	13	0.00
5	9	9	9	0.00	6	6	6	0.00	6	6	6	0.00	5	5	5	0.00	4	4	4	0.00
6	5	5	5	0.00	3	3	3	0.00	3	3	3	0.00	3	3	3	0.00	2	2	2	0.00
7	7	7	7	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00
8	13	13	13.6	0.00	10	10	10	0.00	8	8	8.1	0.00	5	5	5	0.00	5	5	5	0.00
9	8	8	8	0.00	8	8	8	0.00	8	8	8	0.00	5	5	5	0.00	5	5	5.3	0.00
10	8	8	8.1	0.00	5	5	5	0.00	5	5	5	0.00	5	5	5	0.00	5	5	5	0.00

Table 2: Experimental Results II.

	C = 3															
P	$M_{max} = 6$				$M_{max} = 7$				$M_{max} = 8$				$M_{max} = 9$			
	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)
1	27	27	27.8	0.00	18	18	18.6	0.00	11	11	11	0.00	11	11	11	0.00
2	7	7	7	0.00	6	6	6	0.00	6	6	6	0.00	6	6	6	0.00
3	9	9	9	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00
4	27	27	27	0.00	18	18	18	0.00	14	14	14	0.00	13	13	13	0.00
5	11	11	11	0.00	8	8	8	0.00	8	8	8	0.00	6	6	6.1	0.00
6	6	6	6	0.00	4	4	4	0.00	4	4	4	0.00	3	3	3	0.00
7	11	11	11.1	0.00	5	5	5	0.00	5	5	5	0.00	4	4	4	0.00
8	14	14	14	0.00	11	11	11	0.00	11	11	11	0.00	10	10	10	0.00
9	12	12	12	0.00	12	12	12	0.00	8	8	8	0.00	8	8	8	0.00
10	10	10	10.2	0.00	8	8	8	0.00	8	8	8	0.00	5	5	5	0.00

6 Conclusion and future work

In this paper we have presented a new firefly algorithm for solving MCDPs. The metaheuristic is quite simple to implement and can be adapted to binary domains by using specific transfer function and discretization methods. The proposed FA is able to reach 90 of the 90 known global optimums, in which

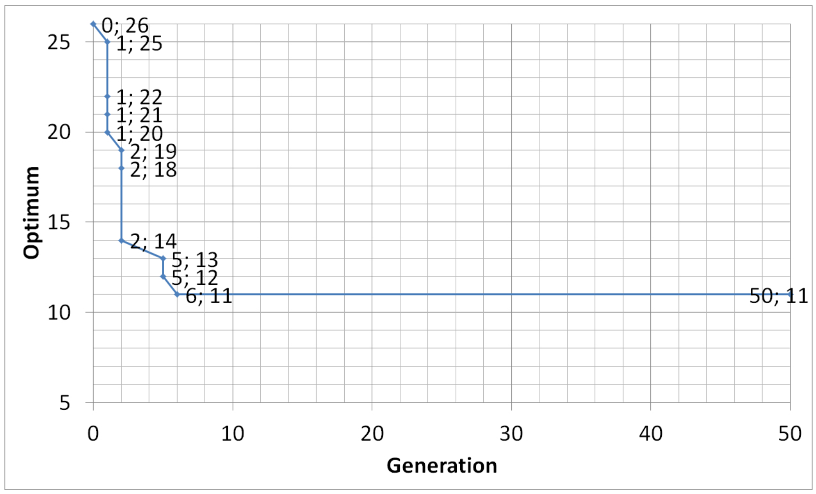

Fig. 1: Performing graph of the FA with C = 2.

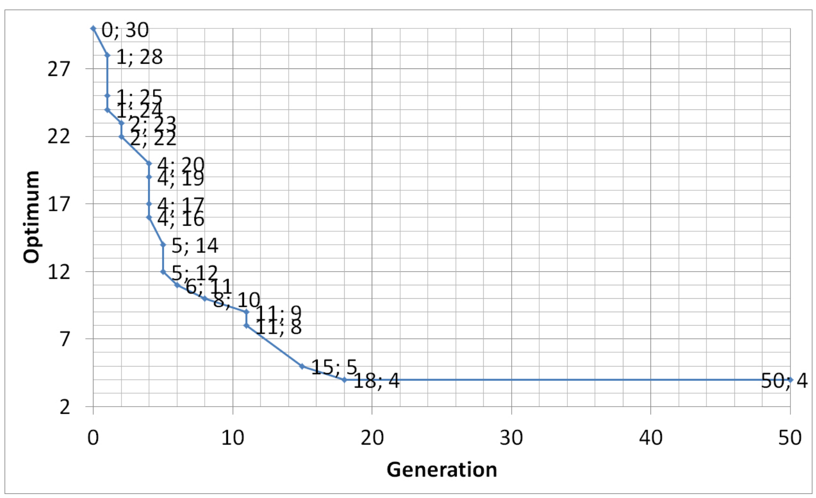

Fig. 2: Performing graph of the FA with C = 3.

runtime per problem turned out to be less than 5 minutes. The results have also exhibited the rapid convergence and robustness of the proposed algorithm which is able to reach reasonable good average global optimums. Indeed, only 11 of 90 problems obtained average values that differ from the global optimum.

As future work, we plan to experiment with additional modern metaheuristic and to provide a larger comparison of modern techniques to solve MCDPs. The integration of adaptive parameter setting to the presented approach would be another direction of research to follow as well.

References

1. G. Adil, D. Rajamani, and D. Strong. A mathematical model for cell formation considering investment and operational costs. *European Journal of Operational Research*, 69(3):330–341, 1993.
2. N. Aljaber, W. Baek, and C.-L. Chen. A tabu search approach to the cell formation problem. *Computers & Industrial Engineering*, 32(1):169–185, 1997.
3. A. Atmani, R. Lashkari, and R. Caron. A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing. *International Journal of Production Research*, 33(1):1–15, 1995.
4. F. F. Boctor. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, 29(2):343–356, 1991.
5. M. Boulif, , and K. Atif. A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. *Computers and Operations Research*, 33(8):2219–2245, 2006.
6. J. Burbidge. Production flow analysis. *Production Engineer*, 42(12):742–752, 1963.
7. K. Chandrasekaran and S. P. Simon. Network and reliability constrained unit commitment problem using binary real coded firefly algorithm. *International Journal of Electrical Power & Energy Systems*, 43(1):921–932, 2012.
8. B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes. A binary firefly algorithm for the set covering problem. In *Proceedings of 3rd Computer Science On-line Conference (CSOC)*, volume 285 of *AISC*, pages 65–73. Springer, 2014.
9. B. Crawford, R. Soto, C. Peña, W. Palma, F. Johnson, and F. Paredes. A binary coded shuffled frog leaping algorithm for the set covering problem. In *Proceedings of the 7th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, volume 9012 of *LNCS*, pages 41–50. Springer, 2015.
10. O. Durán, N. Rodriguez, and L. Consalter. Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. *Expert Systems with Applications*, 37(2):1563–1567, 2010.
11. S. A. Irani. Handbook of cellular manufacturing systems. *John Wiley & Sons*, 1999.
12. T. L. James, E. C. Brown, and K. B. Keeling. A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research*, 34(7):2059–2079, 2007.
13. S. Lozano, A. Díaz, I. Eguía, and L. Onieva. A one-step tabu search algorithm for manufacturing cell design. *Journal of the Operational Research Society*, 50(5):209–516, 1999.
14. E. Oliva-Lopez and G. Purcheck. Load balancing for group technology planning and control. *International Journal of Machine Tool Design and Research*, 19(4):259–274, 1979.
15. G. F. K. Purcheck. A linear-programming method for the combinatorial grouping of an incomplete set. *Journal of Cybernetics*, 5(4):51–78, 1975.
16. S. Sankaran and E. Y. Rodin. Multiple objective decision making approach to cell formation: A goal programming model. *Mathematical and Computer Modelling*, 13(9):71–81, 1990.

17. S. M. Shafer and D. F. Rogers. A goal programming approach to the cell formation problem. *Journal of Operations Management*, 10(1):28–43, 1991.
18. R. Soto, H. Kjellerstrand, O. Durán, B. Crawford, E. Monfroy, and F. Paredes. Cell formation in group technology using constraint programming and boolean satisfiability. *Expert Systems with Applications*, 39(13):11423–11427, 2012.
19. R. L. Storch. Group technology. *College of Engeneering, University of Washington*, 2010.
20. V. Venugopal and T. Narendran. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers & Industrial Engineering*, 22(4):469–480, 1992.
21. T. Wu, C. Chang, and S. Chung. A simulated annealing algorithm for manufacturing cell formation problems. *Expert Systems with Applications: An International Journal*, 34(3):1609–1617, 2008.
22. A. R. Xambre and P. M. Vilarinho. A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European Journal of Operational Research*, 151(2):434–446, 2003.
23. X.-S. Yang. Nature-inspired metaheuristic algorithm. *University of Cambridge, United Kingdom: Luniver Press*, 2008.
24. X.-S. Yang. Firefly algorithms for multimodal optimization. In *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications (SAGA)*, volume 285 of *LNCS*, pages 169–178. Springer, 2009.