

University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Application of multiple objective particle swarm optimisation in the design of damaged offshore mooring systems

Zhuo Wang

University of Wollongong, zw213@uowmail.edu.au

Timothy J. McCarthy

University of Wollongong, timmc@uow.edu.au

M Neaz Sheikh

University of Wollongong, msheikh@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Application of multiple objective particle swarm optimisation in the design of damaged offshore mooring systems

Abstract

The offshore hydrocarbon industry operates in more hostile environments as more of marginal fields become economically viable. This means that more floating production systems and economical mooring systems will be needed. With this increase in the use of marginal fields goes the need to re-use vessels and moorings. Floating production systems, such as FPSO's, need to survive extreme events and extreme damage conditions. When one mooring line is damaged, the remaining ones must be sufficient to avoid a complete failure and still protect critical components such as the riser. This paper looks into applying an evolutionary optimisation technique, namely multiple objective particle swarm optimisation, to the damaged mooring design and analysis. The evaluation of offshore objective functions is computationally expensive since it requires use of complex simulations. When the number of objective function evaluations is large, as is the case with evolutionary methods, even a fast computer takes undesirably long to complete the job. Hence, a robust optimisation algorithm with great efficiency is required to minimise the number of total runs.

Keywords

mooring, offshore, damaged, design, optimisation, systems, swarm, application, particle, objective, multiple

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wang, Z., McCarthy, T. J. & Sheikh, M. Neaz. (2013). Application of multiple objective particle swarm optimisation in the design of damaged offshore mooring systems. *Key Engineering Materials*, 569-570 1257-1264.

Application of multiple objective particle swarm optimisation in the design of damaged offshore mooring systems

Zhuo Wang^{1,a}, Timothy J McCarthy^{1,b} and M Neaz Sheikh^{1,c}

¹ University of Wollongong, Australia

^azhuo@uow.edu.au, ^btimmc@uow.edu.au, ^cmsheikh@uow.edu.au

Keywords: multiple objective particle swarm optimisation (MOPSO), damaged mooring system, mooring design, robust optimisation algorithm

Abstract. The offshore hydrocarbon industry operates in more hostile environments as more of marginal fields become economically viable. This means that more floating production systems and economical mooring systems will be needed. With this increase in the use of marginal fields goes the need to re-use vessels and moorings. Floating production systems, such as FPSO's, need to survive extreme events and extreme damage conditions. When one mooring line is damaged, the remaining ones must be sufficient to avoid a complete failure and still protect critical components such as the riser. This paper looks into applying an evolutionary optimisation technique, namely multiple objective particle swarm optimisation, to the damaged mooring design and analysis. The evaluation of offshore objective functions is computationally expensive since it requires use of complex simulations. When the number of objective function evaluations is large, as is the case with evolutionary methods, even a fast computer takes undesirably long to complete the job. Hence, a robust optimisation algorithm with great efficiency is required to minimise the number of total runs.

Introduction

The design and optimisation of offshore mooring systems is a challenging and laborious task. The difficulties begin with intact mooring design, facilities installation, followed by the practical limitations of line clashing, material fatigue, etc. Before finalising the mooring system, it is necessary to design and optimise the system in the damaged condition. Classical search approaches applied to mooring optimization have limited the scope since they tend to converge to local optimum [1]. Evolutionary design methods have the potential to solve difficult problems in terms of finding global optima with efficiency [2]. Even though some open search evolutionary approaches have a comprehensive ability to find optimum solutions, the complexity of designing damaged mooring systems presents further challenges to find feasible, if not optimum, solutions in many circumstances.

Particle swarm optimisation (PSO) is a heuristic algorithm inspired by the choreography of a swarm of bees. It was first introduced by Kennedy and Eberhart [3]. PSO is based on the premise that individuals share their information within the swarm for the purpose of seeking advantage (goal). According to Kennedy [4], the application of multi-objective PSO (MOPSO) in engineering is particularly suitable due to the speed of its convergence. The use of the evolutionary multi-objective optimisation (EMO) has grown significantly in the last few years, as claimed by Coello Coello et al. [5]. In canonical mooring design and optimisation, one particular objective such as economical cost, is often focused on by engineers, while other concerns are either dealt with as constraints or design variables. This gives engineers the solutions with respect to cost. However, other factors that can potentially improve the performance of the system are neglected. In this paper, the MOPSO is applied to the optimisation of a damaged offshore mooring design problem with the consideration of improving the performance of the whole system.

Particle Swarm Optimisation (PSO)

Like other evolutionary algorithms, particle swarm optimisation (PSO) is based on the population of random solutions, called particles, which are in essence, candidates from the solution space. Each particle has a velocity which allows them to move throughout the domains. The velocity vector is updated through the historical behaviour of the particles. To be specific, the velocities are determined by the history of the best position. There are two types of positions in PSO, the best personal position which is referred to as the best local position, and the best swarm position which is the best global position. The best personal position is the best solution that each individual particle has ever had in its moving trajectory. The best swarm position (sometimes called the best global position) is the best solution selected from all the particles.

Mathematically, the position x of a particle i , at time t is updated as

$$x_t^i = x_{t-1}^i + v_t^i \Delta t$$

(1)

where v_t^i is the velocity vector at time t , and Δt is the time step during two iterations. Fig 1 shows how the particle positions are updated.

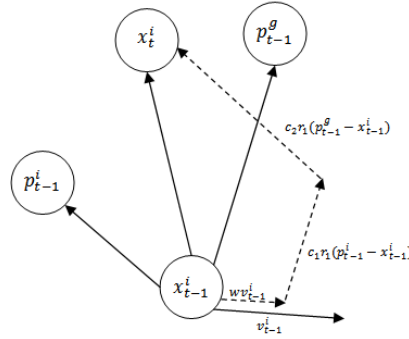


Figure 1. A schematic view of a particle position updates

The ‘time’ incremental step Δt is taken here as a unit for the convenience of calculation. Particles have set values for each of the input variables, which have been defined as the dimensions of the solution space. The velocity component of a particle has been separated into vectors that have multi-dimensions. Each of the dimensions in the velocity represents the changing of a variable. The velocity vector of each particle is calculated as

$$v_t^i = w v_{t-1}^i + \frac{c_1 r_1 (p_{t-1}^i - x_{t-1}^i)}{\Delta t} + \frac{c_2 r_2 (p_{t-1}^g - x_{t-1}^i)}{\Delta t} \quad (2)$$

where both r_1 and r_2 are generated uniformly between 0 and 1 for the purpose of providing randomness; p_{t-1}^i corresponds to the best position for particle i in its time history; p_{t-1}^g represents the best position for all the particles at time $t-1$; c_1 and c_2 are parameters showing the confidence the particle has in itself and among the swarm, respectively; w is the coefficient of inertia.

Multi-Objective Particle Swarm Optimisation (MOPSO)

Chen and Lu [7] claimed that real problems often require an optimisation that considers the nature of several objectives; for instance, purposes such as cost, performance, and reliability are common objectives. Boulougouris [8] confirmed Chen and Lu claim for the multi-objective optimisation when designing an LNG terminal. By setting a single objective function that does not fully represent the design space or feasible solution zone can lead sub-optimal results.

Instead of a single scalar, for which a minimum value can easily be found, a multi-objective algorithm produces a set of good trade off solutions from which decision makers can choose. There

are two main approaches to multi-objective optimisation: namely, aggregation and the Pareto based approach..

The aggregation approach merges the objective function score vector as a utility function by multiplying a weight factor which reflects the relative importance of that objective. This approach allows for the existence of multiple criteria, but its disadvantage is the requirement for weight factors to be tuned after repeated trial and error in order to achieve the desired optimum. Another limitation becomes prominent when there are conflicting constraints. Even strategies such as the one developed by Le Huédé [9] providing trade-offs between criteria by diversifying the search strategies on several branch & bound search iterations are problematic.

Pareto approach

The Pareto approach is based on the concept of Pareto dominance, which has been borrowed from the field of economics. In Pareto dominance, one individual cannot be better off without making another individual worse off. Unlike a single objective PSO, where particles can easily be selected by comparing the scalar objective values, the MOPSO requires the assistance of the more complex Pareto dominance. One of the biggest challenges to MOPSO, according to by Knowles [10], is balancing the diverse maintenance ability and optimum global searching efficiency. MOPSO related works extended PSO to handle multi-objectives on the basis of the Pareto approach. Fieldsend and Singh [11] used a special data structure named ‘dominated tree’ to store the Pareto optimality set for the search process. Apart from the data structure, this approach used a mutation or “turbulence” operator to calculate the velocity. Mostaghim et al [12] improved the convergence and diversity of the MOPSO by adopting a sigma method. This approach resembles the compromise programming proposed by Coello Coello [5].

In recent years the MOPSO techniques [13-14] are more or less related to the use of an external repository that shares particle information to guide their flights. The external repository does not change the main PSO algorithm and only adds an external information pool (or pools) which stores Pareto front information to guide the search trajectories.

According to Coello Coello [14] the main process of applying the external repository in MOPSO have been summarised below:

1. Initialise the number of particles in a swarm
2. Give the initial speed of each particle. This can normally be set as zero
3. Evaluate the objective values of each particle in the swarm
4. Store or memorise the positions of the non-dominated particles in the external repository
5. Generate hyper cubes based on a current Pareto front so far and determine their coordinates according to their objective values
6. Initialise the historic memory of each particle
7. Compute the speed of each particle via

$$v_i^j = wv_{i-1}^j + c_1r_1^j(pbest^j - x_{i-1}^j) + c_2r_2^j(REP[p] - x_{i-1}^j) \quad (3)$$

where the representation of parameters are shown in Eq (2); $pbest^j$ is the best position that the particle i has ever had; $REP[p]$ is a global best particle selected from the external repository introduced lately.

8. Compute the new position using Eq (1)
9. Update the Pareto front Repository
10. Repeat steps 7 to 9 until the stopping criterion be triggered.

When the current position of each particle is better than its previous position in its history, the $pbest^j$ is updated through Eq (3).

$$pbest^j = x^j \quad (4)$$

The criterion of a single objective PSO *pbest* selection is the value of its objective function. While in MOPSO, if the position of the current particle is dominated by its memory position, then the memorised position is kept (case 2 in Fig 2), but if the current position dominates the memorised position, the current position is selected (case 1 in Fig 2). Otherwise, (both of the solution sets are Pareto optimal) the best position is selected randomly (case 3 in Fig 2).

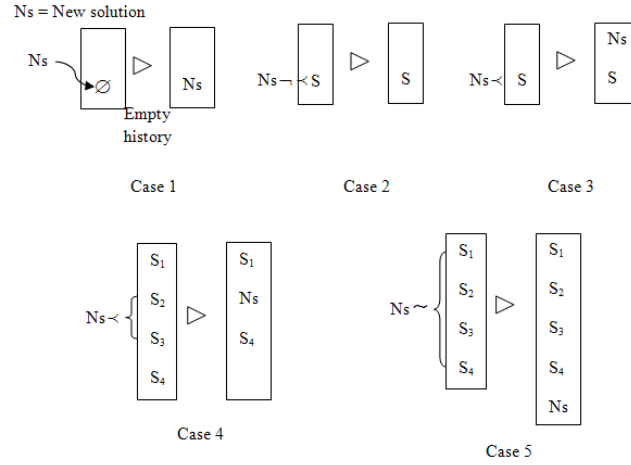


Figure 2. Particle history update diagram

The selection criterion for deciding the best particle position historically is based on the Pareto dominance theory. However, selecting the best particle in a swarm from the repository requires further handling techniques. Coello Coello [14] proposed a hypercube method with a roulette wheel selection. This method forms a grid for the repository through a variation of the adaptive grid proposed by Knowles [10]. The basic idea is to use an external archive to store all the Pareto optimal information with respect to objectives.

The repository building algorithm resembles the process of updating the particle history shown in Fig 2. The difference is that the external repository stores all the Pareto optimal solutions from all the particles in a swarm for the whole iteration. Apart from case 1 to case 3, there are two more cases, as shown in Fig 2, with regards to all the particles. Case 4 shows how the new solution dominates the existing solution in the archive, and how the existing solution is replaced by a new one. Case 5 invokes the adaptive procedure of the external repository. New solutions can be stored in the repository until maximum capacity has been reached.

Design of damaged moorings with MOPSO

This section describes the details of a turret mooring system. The solution and results illustrated here are the design of the mooring system in intact condition. The FPSO is about to be moored in a scenario where LAT is 65 meters. The single turret mooring has a symmetrical 3×3 mooring legs configuration with a fixed anchor radius of 550 meters. Under normal conditions, the offset of the FPSO is limited to 25.5m to accommodate for riser configuration under a fully loaded situation. This offset is assumed to be the result of the excursion from a combination of the dynamic composition of wind, wave, and current in their collinear circumstance. In other words, when the vessel has 25.5m offset, the restoring force provided by the mooring system is in equilibrium with the combined environmental force. For the purpose of the damaged mooring study, the maximum allowable offset of the vessel is increased to 28m. The following concerns are monitored in the process of optimisation.

- The material cost of this mooring system in terms of the volume of steel;
- The maximum amount of tension of the most-loaded mooring leg;

- The maximum requirement of vessel payload reduction;
- The maximum anchor tension format as the other end tension from the most-loaded mooring leg
- The total peak restoring force that can be provided by the mooring system.

MOPSO parameters and settings

Based on the extensive parametric studies of PSO from de Pina [15], the performance of PSO is affected considerably by using the coefficient of inertia weight. Any value less than 0.8 for the inertia weight results in a premature convergence. For inertia weights larger than $w = 0.8$, a steady mean value and low standard deviations are expected. The authors confirmed that the results of PSO were not influenced very much by the cognitive parameter c_1 and social parameter c_2 in SCR design. Perez [6] provided the stability characteristics of PSO through eigenvalue analysis. The inertia weight w is taken as 0.875 in this optimisation software. The cognitive parameter c_1 and social parameter c_2 are taken as 2. The software terminates the execution of PSO at the 100th iteration. The results and comparison of PSO and MOPSO are made in terms of effectiveness and efficiency at the end of the 100th iteration. de Pina [15] used a population of 10 particles in his offshore riser optimisation.

The coefficients for the PSO used in the current study are presented in Table 1.

Table 1. PSO and MOPSO experiment coefficients

Particle number in a swarm	10
Number of iterations	100
Cognitive parameter (c_1)	2
Social parameter (c_2)	2
Inertia weight (w)	0.875

Design variables

Figure 3 presents a plan view of an intact turret mooring model showing the parameters of design variables that are considered for the mooring optimisation. The FPSO is about to be moored in a scenario where LAT is 65 meters. The optimisation is taken with one line damaged condition. As shown in Fig 3, the most loaded mooring line from the intact analysis is leg number 9. In the damaged design, it is assumed that the mooring leg 9 is damaged. For simplicity of the optimisation, it is assumed that rest of the legs have the same material and geometrical properties. Every anchor leg comprises an upper chain segment (L_1), a middle chain segment (L_2) and a ground chain segment (L_3). The total length of a single mooring cable is indicated as ($L_1 + L_2 + L_3$). There are also other parameters related to the mooring cable, which are the diameter of the upper chain segment (D_1), diameter of the middle chain segment (D_2) and diameter of the ground chain segment (D_3). To be offshore engineering practical, the chain diameters are arbitrarily selected from the Ramnäs product manual [16]. The selection of chain diameters is based on 33 indices of the chain classifications that range from 70mm to 177mm.

The main design variables in the optimisation problem are L_1 , L_2 , L_3 , D_1 , D_2 , D_3 , and their design limits are listed in Table 2. The values of other related variables are calculated from those parameters accordingly.

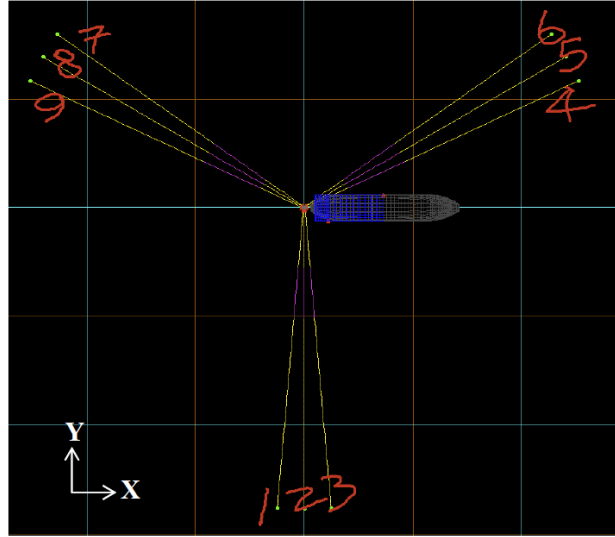


Figure 3. Plan view of the intact mooring system

Table 2. Limits range of segment lengths

	<i>Minimum</i>	<i>Maximum</i>
<i>Top segment length (L_1)</i>	114 m	130 m
<i>Middle segment length (L_2)</i>	88 m	105 m
<i>Ground segment length (L_3)</i>	348 m	365 m
<i>Top segment diameter (D_1)</i>	70 mm (Index 1)	177 mm (Index 33)
<i>Middle segment diameter (D_2)</i>	70 mm (Index 1)	177 mm (Index 33)
<i>Ground segment diameter (D_3)</i>	70 mm (Index 1)	177 mm (Index 33)

Objective functions

It is common in engineering optimisation problems that the objective function is associated with the lowest cost. The compliance of other conditions is defined in the penalty functions. For this damaged mooring problem, the cost function can simply be taken as the cost of materials. Therefore, the cost function of the mooring problem is given by:

$$f = \sum_{i=1}^8 \left(\frac{\pi D_{e,i}^2}{4} \times L_i + P_{B,i} + P_{R,i} + P_{P,i} + P_{D,i} \right) \quad (5)$$

where D_e is the equivalent diameter of the cable/chain, L is the length of the cable, P_B is the value of penalty function due to cable breaking load violations, P_R is the value of the penalty function corresponding to the restoring forces violations by the mooring system, P_P is the value of the penalty function as to payload violations, and P_D is the penalty value as per anchor tension violations. The penalty functions are introduced in the following section.

MOPSO has the ability to accommodate two or more objectives. To demonstrate its performance in this case study, two objectives were chosen. The first objective is the cost of the mooring legs, which is to minimise the cost of materials. The second objective can be selected from the concerns listed. The most frequently seen violations were either an inadequate restoring force to withstand the environment or insufficient anchor tension from intact analysis. Therefore, the second objective adopted for optimisation is selected as the restoring force that the damaged mooring system can provide.

Penalty functions

A penalty function associated with a violation of the constraints is defined as a ratio of x between the simulated values and the constraint limits, as follows:

$$P = \begin{cases} \sum k \times (x), & \text{if } x \geq 1 \\ 0, & \text{if } x < 1 \end{cases} \quad (6)$$

where k is the coefficient associated with each criterion. The factor k is the penalty threshold that is the initial penalty when any constraints have been violated. It was applied as soon as the penalty scheme had been triggered. It is a quantitative criterion that distinguishes the boundary between feasible and constraints-violated particles. The value of the k factor is associated with the tolerance of emerging non- constrained solutions.

Results and discussion

For the first objective, the goal is to minimise as much as possible. While the second objective, the restoring force has to be capable of being greater than or equal to the external environmental force.

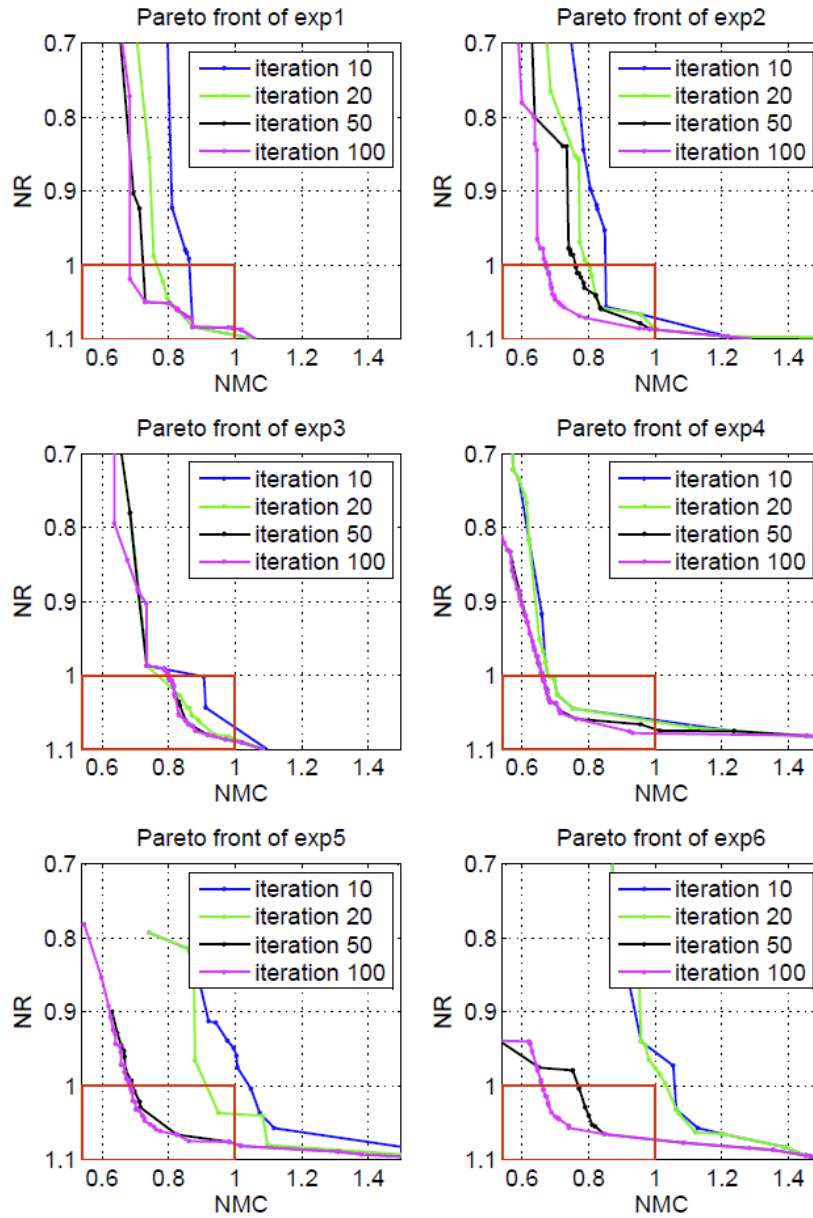


Figure 4. Pareto front curves with different iteration numbers

The normalised results are plotted in Fig.4 where NMC stands for normalised material cost and NR means normalised restoring force. The convergence of the MOPSO can be evaluated with the

Pareto front curve (as shown in Fig.4). The rigid red rectangle in the diagram represents the improvement zone, which means any particles located in this region are better than the reference mooring system. It is clear that the Pareto front curve is approaching the improvement zone with the increasing iterations. In other words, the particles are guided to the expected directions with the increase of iterations. Even though the performance is improving as the iterations increase, it is unwise, computationally, to let the algorithm run forever.

Summary

This work has demonstrated the ability of MOPSO in designing and optimising damaged offshore mooring systems. The results found that with the aid of MOPSO, the searching ability of the algorithm has been enhanced. Meanwhile, chances of finding optimised solutions have increased. However, since this study was only carried in damaged mooring design in its static situation, a full study and comparison in its intact and dynamic analysis is recommended for future work.

References

- [1] Z. Wang, T J McCarthy and N Sheikh, "A comparison of Genetic Algorithm and Particle Swarm Optimisation for Theoretical and Structural Applications," in Proceedings of the Eleventh International Conference on Computational Structures Technology, Dubrovnik 4-7 Sep 2012, Paper 68, Civil-Comp Press, Stirlingshire, UK doi:10.4203/ccp.99.68, 2012.
- [2] N. Cunliffe, C. Baxter, and T. McCarthy, Evolutionary design of marine riser systems, in 23rd International conference on offshore mechanics and arctic engineering. 2004: Vancouver, Canada
- [3] J. Kennedy, and R. Eberhart, Particle swarm optimization. IEEE international conference on neural networks, 1995. IV: p. 1942-8.
- [4] J. Kennedy, and R.C. Eberhart, Swarm Intelligence. 2001, San Mateo, CA: Morgan Kaufmann
- [5] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamout, Evolutionary Algorithms for Solving Multi-Objective Problems. 2002: Norwell, MA: Kluwer.
- [6] R.E. Perez, and K. Behdinan, Particle swarm approach for structural design optimization. Computers and Structures, 2007. 85(2007): p. 1579-88.
- [7] M.R. Chen, and Y.Z. Lu, A novel elitist multiobjective optimisation algorithm: Multiobjective extremal optimisation. European Journal of Operational Research, 2008. 188: p. 657-51.
- [8] E.K. Boulougouris, and A.D. Papanikolaou, Multi-objective optimisation of a floating LNG terminal. Ocean engineering, 2008. 35: p. 787-811
- [9] F. Le Huédé, et al., MCS - a new algorithm for multicriteria optimisation in constraint programming. Ann Oper Res, 2006. 147: p. 143-74
- [10] J.D. Knowles, and D.W. Corne, Approximating the nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary computation, 2000. 8(2): p. 149-72.
- [11] J.E. Fieldsend, and S. Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. in Proc. 2002 U.K. Workshop on Computational Intelligence. 2002. Birmingham, U.K.
- [12] S. Mostaghim, and J. Teich. Strategies for finding good local guides in Multi-Objective Particle Swarm Optimisation (MOPSO). in Proc. 2003 IEEE Swarm Intelligence Symp. 2003. Indianapolis,
- [13] P.K. Tripathi, S. Bandyopadhyay, and S.K. Pal, Multi-Objective Particle Swarm Optimisation with time variant inertia and acceleration coefficients. Information Sciences, 2007. 177: p. 5033-49.
- [14] C.A. Coello Coello, G.T. Pulido, and M.S. Lechuga, Handling Multiple Objectives with Particle Swarm Optimisation. Evolutionary computation, 2004. 8(3): p. 256-79.
- [15] de Pina, A.A., B.S.L.P. de Lima, and B.P. Jacob, Tailoring the particle swarm optimization algorithm for the design of offshore oil production risers. Optimization and engineering, 2011. 12: p. 215-35.

[16] <http://ramnas.com/wp-content/uploads/2012/11/Ramnäs-Product-Catalogue2.pdf> (visited 14 Feb 2013)