24th Sept 2017

Trade compression as a linear optimization problem

Robert Sipos (PhD) and Viktor Nagy (PhD)

Talk Summary

- Introducing trade compression as a concept.
- Financial, regulatory motivation.
- Simple overview of algorithms and approaches used in the industry, introducing a naive loop algorithm.
- Finding an optimal solution: trade compression as a linear optimization problem.

Trade compression in the financial industry

FINANCIAL TIMES

myFT

HOME WORLD US COMPANIES MARKETS OPINION WORK & CAREERS LIFE & ARTS Sign In Subscribe

Subscribe to the FT to read:

"Compression of swap trades gains pace as banks tear up old trades"

The Economist

World politics Business & finance Economic

Trade compression

Number crunch

The CDS market has shrunk dramatically. Why?

Jul 9th 2009

Timekeeper

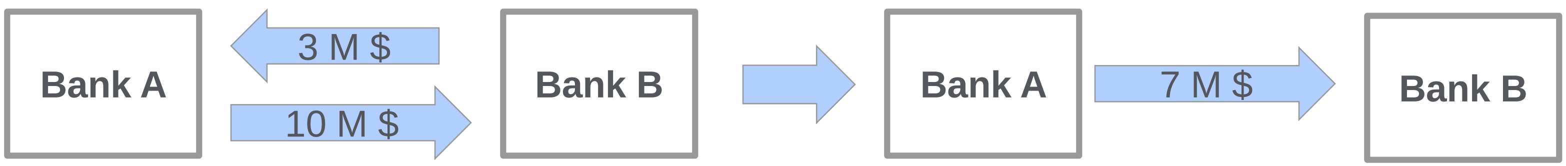
The size of the credit-default-swap (CDS) market, which stood at a monumental $57 trillion dollars in June 2008, has grabbed attention. That number has since shrunk dramatically. According to the Bank for International Settlements, the value of outstanding CDS contracts had fallen to $42 trillion in December.

The apparent collapse is largely down to something called trade compression.

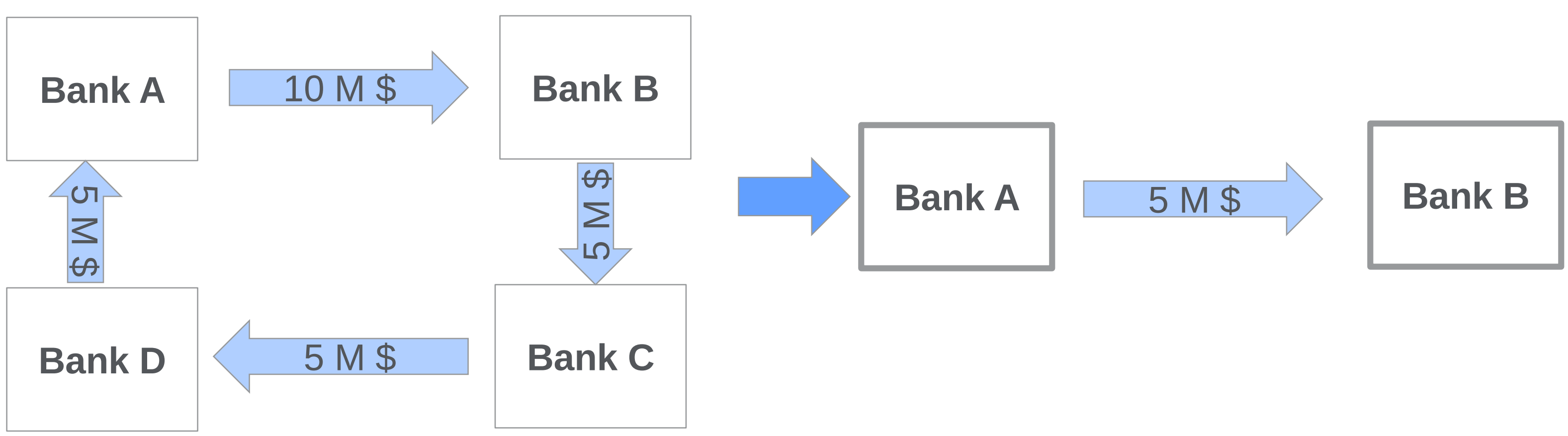
My first guess…

What is trade compression?

- Trade compression is used to reduce the number of contracts that banks have on their books, while keeping the same economic exposure (present value, risks).
- *Unilateral basis*: cancelling offsetting contracts in their own portfolio.
- E.g., *bilateral basis*, firms cancel offsetting involving two parties:

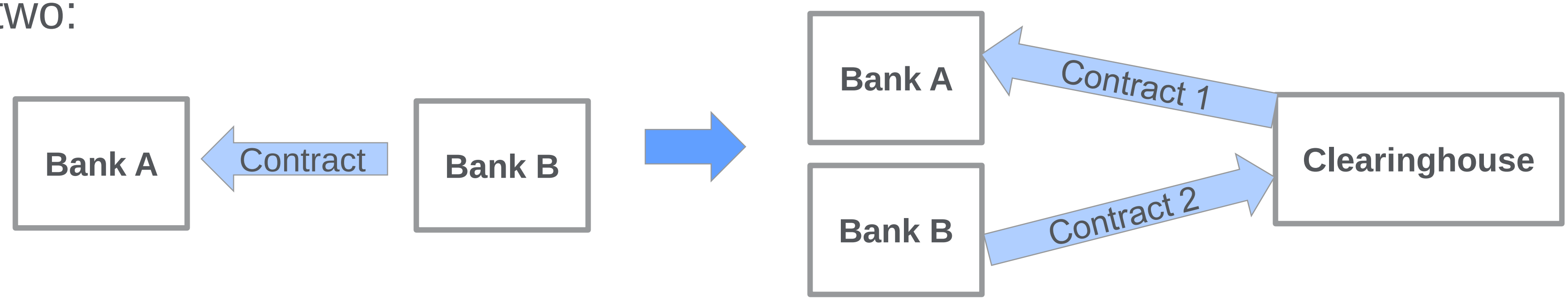


- Multilateral *basis*, firms cancel offsetting contracts involving multiple parties:



What is the economic motivation for performing compressions?

- Reducing number of outstanding contracts (number of trades, total notional).
- Reduction of capital needed to cover trading book risk.
- Mitigating credit/counterparty risk.
- Easier to manage trading book and hedge with a smaller number of positions.
- There are firms offering algorithms for compressing trades: TriOptima and CLS are the main market providers.
- The issue is even more acute with central clearing, as one position turns into two:



Trade compression

- Compression is an optimization problem, where the task is to find a portfolio for each counterparty that is equivalent to the original one, while minimalizing a measure (e.g., number of trades). Equivalence could be defined with a set of boundary conditions:
  - Present Value
  - Credit Risk
  - Total Notional

  Can be prescribed to remain constant.

- Compression effectiveness in greatly facilitated by the level of standardization of the underlying product. For instance, if maturities, coupons are standardized, as it is the case for CDS, then finding offsetting positions is more straightforward.

Example: compressing CDS (insurance against default trades)

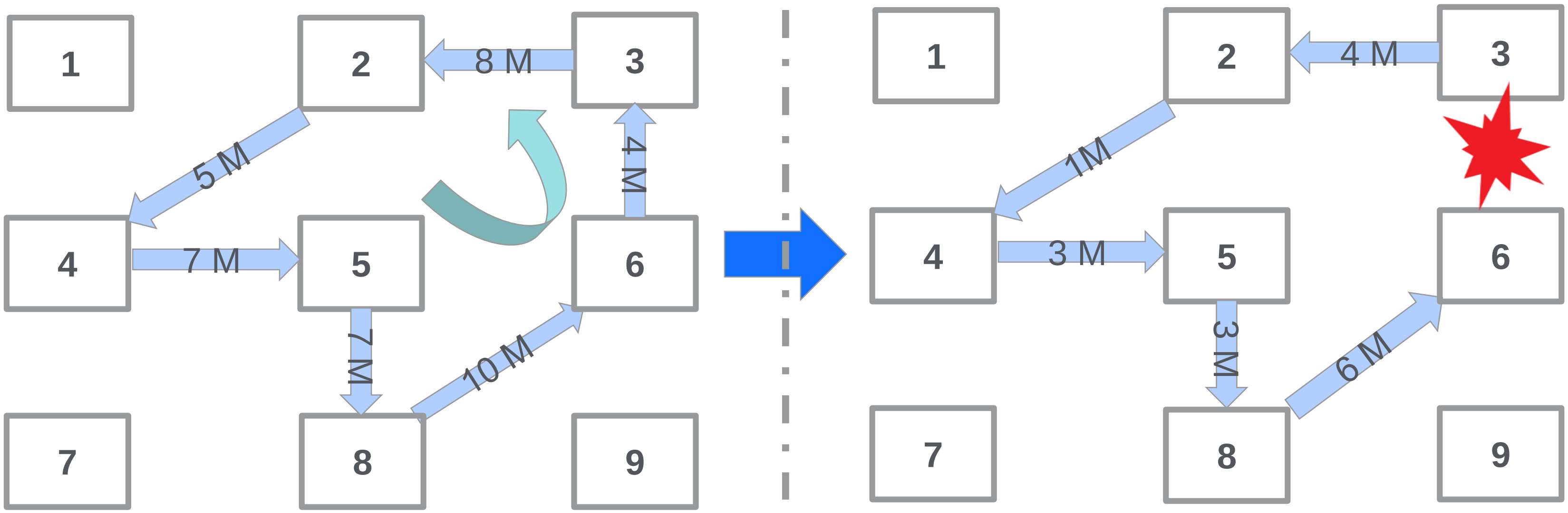
- We assume that we have N counterparties.
- We have one generic type of contract a CDS that is an insurance against default, like car/home insurance. I.e., paying a quarterly premium and get reimbursed if a default event happens (car collision).
- We assume that counterparty (i) bought insurance with a notional $C_{i,j}$ on a reference entity from every other counterparty (j) .
- $C_{i,j}$ is antisymmetric i.e., $C_{i,j} = -C_{j,i}$, because buying protection for one counterparty means selling for another. Also naturally $C_{i,i} = 0$.
- $\begin{bmatrix} 0 & \cdots & C_{1,N} \\ \vdots & \ddots & \vdots \\ -C_{1,N} & \cdots & 0 \end{bmatrix}$ net long positions for each: $h_i = \sum_j C_{i,j}$

Example: compressing CDS, (naive algorithm).

- The goal is to minimize the counterparty exposure measure $L_2(C) = \sum_{i,j} C_{i,j}^2$, while keeping the exposure $h_i = \sum_j C_{i,j}$ for each counterparty the same.

Loop Compressing Algorithm (Trioptima):

- Enumerate all closed loops (e.g., depth first search algorithm).
- Net each loop, subtract the smallest element from each at once.



Example: compressing CDS, part III.

- This algorithm will reduce $L_2(C)$, while keeping the exposures constant: h_i.
- Netting applies only to closed loops, the number of these grows exponentially and could become very expensive.
- It is not clear how we could incorporate additional conditions (on top of exposure preservation).
- Not a global optimal algorithm.
- One can find optimal compression algorithms for $L_1(C)$ using linear programming for this problem. A simple unilateral compression algorithm will be detailed in the second half of this presentation.

Example

Trade compression in an unilateral case

Example of trade compression in an unilateral case

- Until now we have looked at cases involving multiple parties, now we are trying to compress trades in a single portfolio.
- This example was implemented in our current production system.
- The goal is to reduce the number of securities in a portfolio to minimize the computational cost for our simulations, while keeping the additive risk profiles constant.
- Unfortunately, solving this is a NP-complete problem. Instead, we investigate a related question that approximates this: Minimalizing the total absolute notional $L_1(C)$.
- The new portfolio will be a linear combination of the original.
- Such additive risks are:
  - Current Market Price (MTM) of the securities.
  - First and higher order derivatives of the market price.

Compression for one counterparty

- We have N-trades in a portfolio.
- Let's denote the vector of nationals corresponding to these by $[\mathrm{c}_1 \cdots \mathrm{c}_\mathrm{n}]$.
- We have M-risk factors (j) for each trade (i): $\boldsymbol{A_{i,j}}$. E.g., Market Price (MtM) and it's derivatives. The sum of these risk factors over the whole portfolio is:

$$[\boldsymbol{\Sigma_1} \quad \dots \quad \boldsymbol{\Sigma_M}]$$

- We organize this information as follows: $\begin{bmatrix} \boldsymbol{c_1} & \boldsymbol{A_{1,1}} & \dots & \boldsymbol{A_{1,M}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{c_N} & \boldsymbol{A_{N,1}} & \dots & \boldsymbol{A_{N,M}} \end{bmatrix}$

We seek a vector of portfolio weights $\begin{bmatrix} \boldsymbol{w_1} \\ \vdots \\ \boldsymbol{w_N} \end{bmatrix}$ with the boundary condition:

$$\mathbf{0} \leq \boldsymbol{w_i} \leq \boldsymbol{w_{max}}$$
$$\boldsymbol{\Sigma} + \mathbf{L} \leq \mathbf{A}^T\mathbf{w} \leq \boldsymbol{\Sigma} + \mathbf{U}$$

Where $\mathbf{L}$ and $\mathbf{U}$ are tolerance levels.

Problem as linear programming

- Minimalizing the total notional under boundary conditions:

$$\mathbf{w}: \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbf{c}^T \mathbf{w}$$

such that
$$\mathbf{0} \le w_i \le w_{max}$$
$$\boldsymbol{\Sigma} + \mathbf{L} \le \mathbf{A}^T\mathbf{w} \le \boldsymbol{\Sigma} + \mathbf{U} \longrightarrow \begin{bmatrix} \mathbf{A}^T\mathbf{w} \\ -\mathbf{A}^T\mathbf{w} \end{bmatrix} \le \begin{bmatrix} \boldsymbol{\Sigma} + \mathbf{U} \\ -(\boldsymbol{\Sigma} + \mathbf{L}) \end{bmatrix}$$

canonical form

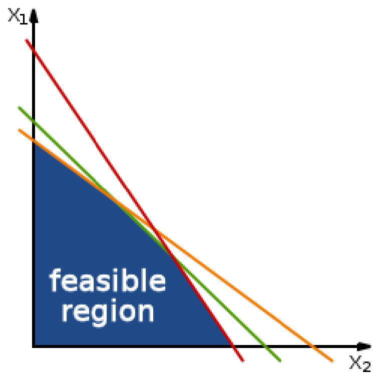
→ LP optimization problem

- Initial solution: $\mathbf{w} = \mathbf{1}$ (not optimal, but feasible)
- Solution is not unique
- Solution tends to be sparse

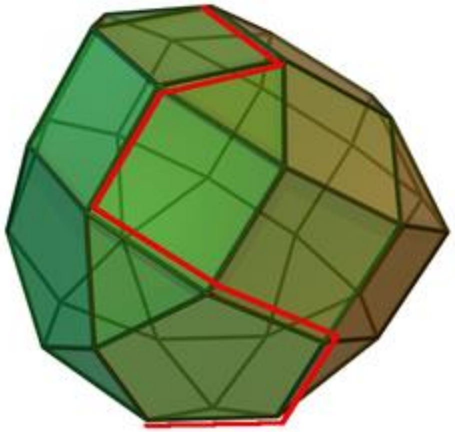
How to solve LP problems – Simplex algorithm

- Graphical method

2-D



N-D



- Construct the smallest convex hull based on the constraints
- Walk on the edges towards increasing/decreasing objective function (greedy algorithm)
- (Too many vertices for exhaustive search.)
- Global optimum is guaranteed
- Usually efficient, worst-case exponential
- Listed as one of the top 10 algorithms in the 20th century (by IEEE)

How to solve LP problems – Other methods

- Other group of algorithms: interior point methods
  - Can visit any point in the feasible region, not only vertices
  - Similar efficiency as the simplex algorithm, depends on the actual problem
- Approximate algorithms: solution is $O(1+\varepsilon)$ optimal [1]
  - Matrix $\mathbf{A}$ is n x m, and has N non-zero entries
  - Best sequential takes $O(N + (logN)(n+m)\varepsilon^{-2})$ time
  - Best parallel takes $O\left((logN)^2\varepsilon^{-3}\right)$ time

[1] Allen-Zhu, Zeyuan, and Lorenzo Orecchia. "Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel." In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1439-1456. Society for Industrial and Applied Mathematics, 2015.

Implementation

- Using NAG
  - Leading numerical algorithm library
  - Efficient and well maintained functionality
  - `nag_opt_lp (e04mfc)`
- <u>Example:</u>

Original:	266 trades	754 bn INR notional
Compressed:	26 trades	88 bn INR notional
	9.77%	**11.69%**

Summary

- Trade compression is therefore an important means of reducing gross notional amounts
  - For achieving regulatory capital savings; and reducing operational and counterparty risk exposures.
  - When used on a multilateral basis, for example, with cleared OTC derivatives trades, it also cuts back on double counting of risk.
  - The unilateral trade compression saved a lot of computational resources when calculating CVA.
  - Therefore the importance of trade compression should not be underestimated.

References:

- **Dominic O’Kane: Optimizing the compression cycle**